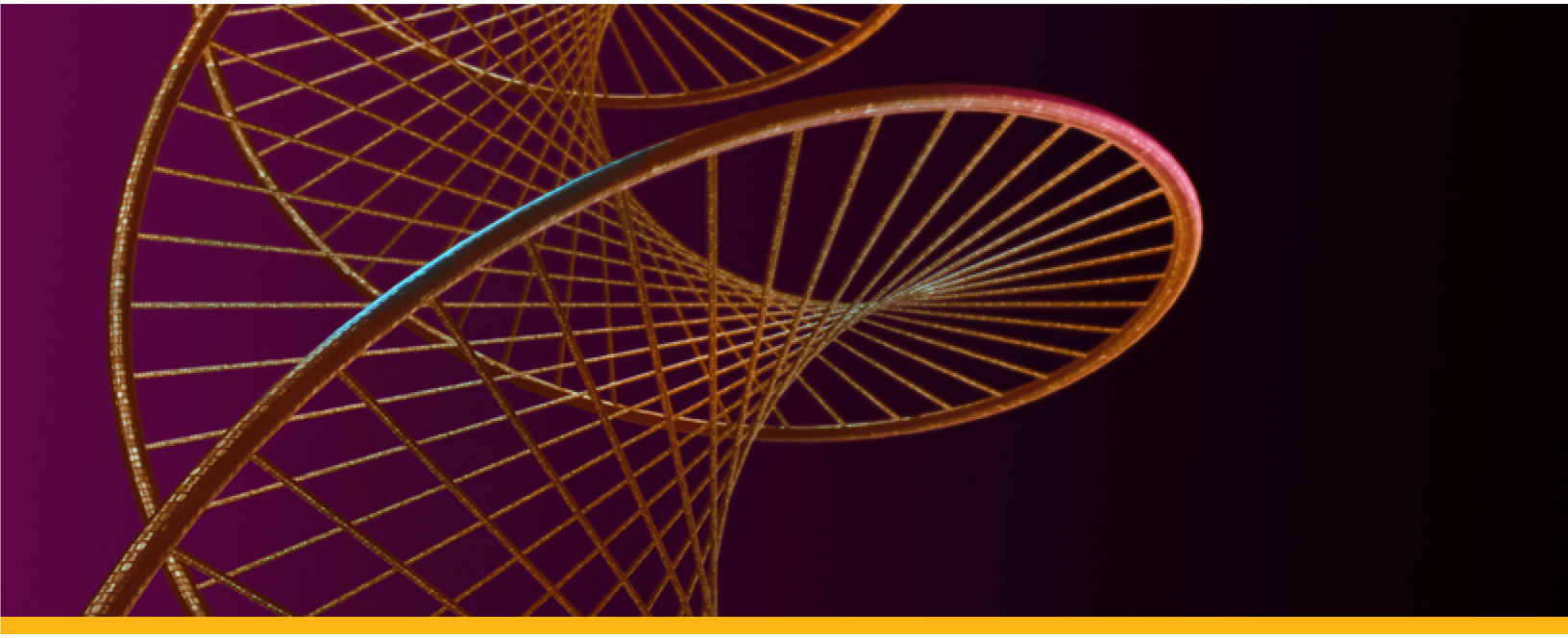
# CyberArk Trust Protection Foundation 26.1

Certificate Authority and Hosting Platform Integration Guide



CYBERARK®

## Legal Notices



We welcome your feedback on our documentation. If you have something you'd like to share, or a way we could improve, please send the following information to documentationfeedback@paloaltonetworks.com: [1] The name of the PDF guide (Certificate Authority and Hosting Platform Integration Guide); [2] The version of CyberArk Trust Protection Foundation - Self-Hosted you are using (26.1); [3] Your feedback.

## Online Documentation

This manual and other CyberArk documentation are available at the CyberArk Documentation Portal at https://docs.cyberark.com/mis-self-hosted/26.1.

## CyberArk Customer Support

| | |
|---|---|
| **Phone:** | Americas: +1-855-636-1536 |
| | EMEA: +44-800-0668865 |
| | APJ: +65-3158-6603 |
| | Israel: +972-1809-344-251 |
| | Australia: +61-1-800-867-206 |
| | India: 000800-919-1790 |
| **Email:** | support@cyberark.com |
| **URL:** | https://community.cyberark.com |

Released: July 2026. Document Version Number: 26.1.0

# About This Guide

Welcome to CyberArk Trust Protection Foundation™. Trust Protection Foundation helps you to rapidly develop an accurate certificate and SSH key inventory and identify security and operational risks. Additionally, you can quickly evaluate your organization's compliance with corporate and regulatory folders and establish a concise methodology to ensure compliance. With built-in management and policy best practices, Trust Protection Foundation helps eliminate data breaches, security audit failures and unplanned system outages.

The *Certificate Authority and Hosting Platform Integration Guide* presents the information you need to successfully integrate CAs and various hosting platforms by configuring and managing CA templates and application drivers.

## Audience

This guide is intended for those responsible with integrating certificate authorities (CAs) with various hosting platforms and the CyberArk Trust Protection Foundation™.

## Terminology

For definitions of terms used in Trust Protection Foundation documentation, refer to the Glossary found in the *CyberArk Trust Protection Foundation Overview Guide*.

# 1

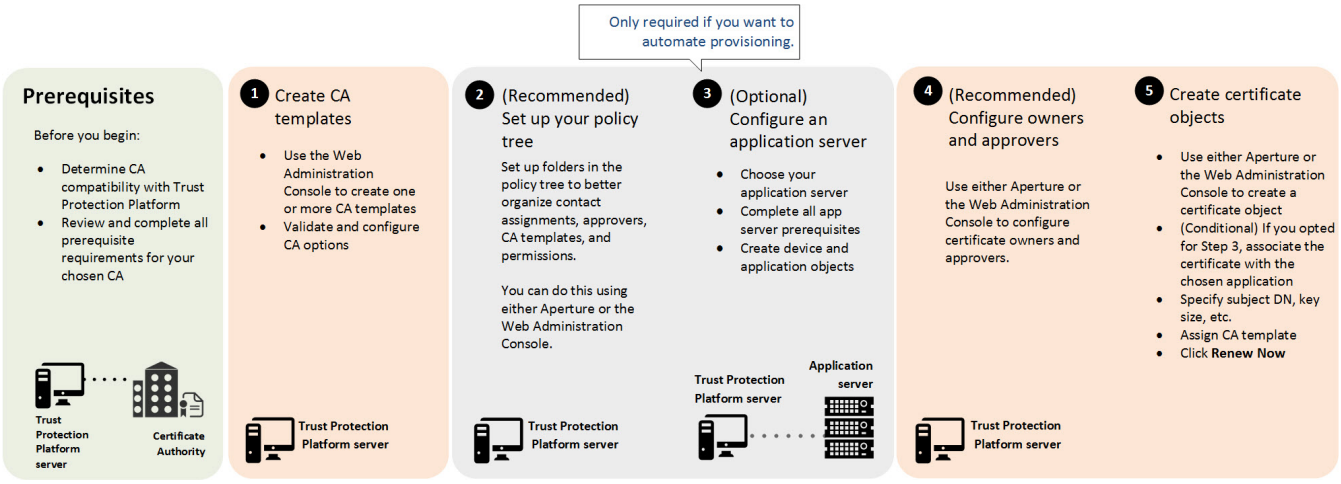
# Getting started: automating certificate enrollment and provisioning

By connecting Trust Protection Foundation with your certificate authority (CA), you can automate the process of creating, renewing, enrolling and provisioning certificates.

Automation is a multi-step process, depending on whether or not you need Trust Protection Foundation to install certificates on application servers.

> **TIP** Even if you've already connected Trust Protection Foundation with your CA, you might want to review this section to verify that you have set things up correctly.

The following diagram shows the end-to-end process for connecting Trust Protection Foundation with your CA or application server to automate certificate provisioning.




*Prerequisites*: Before you begin configuration tasks, review prerequisite information. See "About CA driver prerequisites" on the facing page.

1. *Create CA templates*: Use CyberArk's Policy Tree to create CA templates for your chosen certificate authority. See "Step 1: Create CA templates" on page 22.

2. *(Recommended)Set up your policy tree*: Although using policies is optional, you'll find that doing so makes managing your certificates, CA templates, applications, and devices easier in the long run. See "Step 2: (Recommended) Set up your policy tree" on page 24.

3. *(Optional)Configure an application server*: If you are using an application server, carefully review it's prerequisites and related documentation and then create device and application objects on the Trust Protection Foundation server. See "Step 3: (Optional) Configure an application server" on page 26.

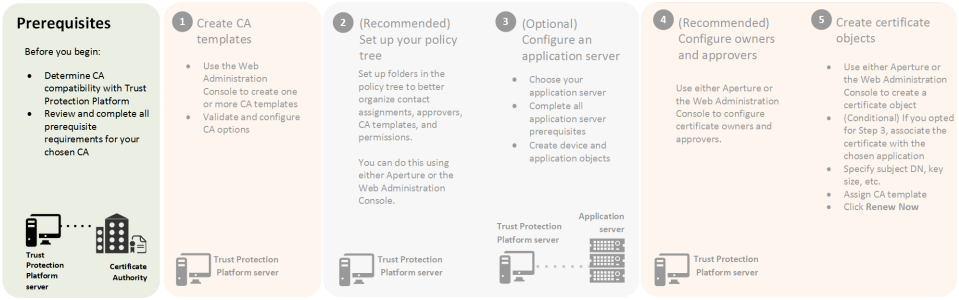
4. *(Recommended)Configure owners and approvers*: Determine who in your organization will own and approve the certificates. See "Step 4: (Recommended) Configure contacts and approvers" on page 27.

5. *Create certificate objects*: With all of the pieces in place, Trust Protection Foundation can automate the issuance of your certificates and install them wherever they are needed. See "Step 5: Create certificate objects" on page 28.

This chapter contains the following topics:

# About CA driver prerequisites

Before you attempt to configure a CA template, you should gather the required information and perform any relevant tasks.




## Determine CA compatibility with Trust Protection Foundation

The CyberArkDrivers Library includes several CA drivers you use to connect Trust Protection Foundation with your CA.

Review the CyberArk Drivers Library to see if a driver already exists for your chosen CA. For a list of CyberArk-supported CAs, see "Trust Protection Foundation enrollment and revocation driver support" on page 32.

If a driver does not exist for your CA (or if an existing CA driver requires additional data fields), you can use the Adaptable CA driver. For more information, see "Adaptable CA" on page 61.

## Review and complete all prerequisite requirements for your chosen CA

Before you start setting up integrations with your CA, make sure that you do the following first:

- Verify network connectivity to your chosen CA.

  Confirming that the CA's web service is reachable helps to ensure that the Trust Protection Foundation server can also reach that service. Typically, this should have been done before you install Trust Protection Foundation. Use whatever tool or method you want to use to send a simple request to your CA's web service. For example, you could use the Windows command line utility and send a `ping` request.

- Using your CA administration tool, grant the *necessary permissions* at the CA to the account that will be used by Trust Protection Foundation.

  Refer to the documentation provided by your CA to configure permissions.

- Update your firewall rules to allow the Trust Protection Foundation server to connect to the CA web service.

  This might require working with another administrator in your organization to modify current firewall rules.

- Configure the necessary credentials for Trust Protection Foundation to connect with your CA's web services.

  Trust Protection Foundation must be able to authenticate with your CA in order to automate certificate work. So you must provide the necessary credentials for your CA. You specify these credentials during "Step 1: Create CA templates" on page 22.

# Supported integrations with Trust Protection Foundation

This topic lists all types of integrations–CAs, devices, applications, services, etc.–that are supported by CyberArk Trust Protection Foundation.

| Vendor | Supported Products | Supported Versions* | Integration Types |
|---|---|---|---|
| Amazon | Amazon Web Services (AWS) | ACM, IAM, ALB, ELB, CloudFront | Cloud Service |
| Amazon | Amazon Certificate Manager (ACM) | | Certificate Authority |

| Vendor | Supported Products | Supported Versions* | Integration Types |
|---|---|---|---|
| Apache | HTTP Server | 2.2 and 2.4 | Application |
| Bouncy Castle | Clients using BC bcpkix library | 1.64 | Certificate Enrollment via EST Protocol |
| Cisco | IOS | 15.7(3)M3 | Certificate Enrollment via EST Protocol |
| Cisco | libest (Client) | 1.1.0 | Certificate Enrollment via EST Protocol |
| Citrix | NetScaler VPX | 14.1, 13.1 build 42.47 | Network Appliance |
| Compuware | VAM w/nShield 6000e F2 | V11.05.1245 (64 bit) | Network Appliance |
| CyberArk | Enterprise Password Vault | 12, 14.2, 14.6 | Credential Provider |
| Dell | iDRAC 8 (using RACADM 8.3)** | 2.41.40.40 firmware | IoT |
| DigiCert | DigiCert CertCentral | NA | Certificate Authority |
| Entrust nShield | Entrust nShield Connect HSM | 12.40.2 (client; minimum version) | Hardware Security Module |
| F5 | Big-IP Local Traffic Manager (LTM) / Application Delivery Controller (ADC) | 21.0.9.1 build 0.0.5, 16.1.3.5 build 0.0.5, 17.1.0.2 build 0.0.2, 17.5.0 Build 0.0.15 | Network Appliance |
| GeoTrust | GeoTrust Reseller | NA | Certificate Authority |
| GlobalSign | GlobalSign MSSL | NA | Certificate Authority |

| Vendor | Supported Products | Supported Versions* | Integration Types |
|---|---|---|---|
| Hewlet-Packard (HP) | iLO 4 (using HPQLOCFG 1.5)** | 2.50 firmware | IoT |
| HID PKIaaS | HID PKIaaS | NA | Certificate Authority |
| IBM | WebSphere DataPower IDG | 10.5.0, 10.6.0 | Network Appliance |
| IBM | TeaLeaf Passive Capture Application (PCA) | v2 3400 | Appliance |
| IBM | Sterling Connect:Direct with Secure+ | 6.0x, 6.1x, 6.2x, 6.3x, 6.4x for Windows and 6.0x, 6.1x, 6.2x, 6.3x, 6.4x for UNIX | Application |
| IBM | Global Security Kit (GSK) | 7.0.3.15, 7.0.4.20 (gsk7cmd, gsk7capicmd & iKeyMan); 8.0.14.34 (gsk8capicmd) | Keystore |
| Imperva | MX | 13.6 | Application |
| Microsoft | Active Directory Certificate Services (ADCS) | Enterprise or Standalone CA running on Windows Server 2003-2012, 2016, 2019, and 2022 | Certificate Authority |
| Microsoft | Internet Information Services (IIS) | 8.0, 8.5, 10.0.1607, and 10.0.1809 | Application |
| Microsoft | Azure | Key Vault, Web App | Cloud Service |
| Microsoft | Windows Certificate Store (CAPI) | Windows Server 2012, 2012 R2, 2016, and 2019 | Keystore |
| Mozilla | Network Security Services (NSS)<br><br>For more information, visit https://en.wikipedia.org/wiki/Network_Security_Services. | 3.x | Keystore |
| OpenSSL | PEM | NA | Keystore |
| OpenSSL | OpenSSL CA | 1.0.0, 3.0.0 | Certificate Authority |

| Vendor | Supported Products | Supported Versions* | Integration Types |
|---|---|---|---|
| OpenTrust | Enterprise PKI | 4.7.1 | Certificate Authority |
| Oracle | Java Keystore (JKS and JCEKS) | 1.6, 1.7, 1.8 | Keystore |
| Oracle | Sun Java System Web Server / Oracle iPlanet Web Server<br><br>For more information, visit https://en.wikipedia.org/wiki/Oracle_iPlanet_Web_Server. | 6.1, 7.0 | Application |
| Palo Alto Networks | Next Gen Firewall | 11.1.*x*+, 11.2, 12.1 | Appliance |
| QuoVadis | QuoVadis | NA | Certificate Authority |
| RedHat | Red Hat Certificate System | 8.1 | Certificate Authority |
| Riverbed | SteelHead WAN Optimizer | VCX555H 9.0.0b | Appliance |
| RSA Security | RSA Certificate Manager | 6.8 and 6.9 | Certificate Authority |
| RSA Security | PKCS#12 | NA | Keystore |
| Sectigo | Sectigo Certificate Manager (CCM) | NA | Certificate Authority |
| Symantec | Symantec Managed PKI for SSL | NA | Certificate Authority |
| Symantec | DigiCert PKI Platform*** | NA | Certificate Authority |
| Thales | estclient | 1.0.1 | Certificate Enrollment via EST Protocol |
| Thales | SafeNet Luna SA | 6.22 (client) | Hardware Security Module |

| Vendor | Supported Products | Supported Versions* | Integration Types |
|---|---|---|---|
| Verizon | CyberTrust UniCERT | 5.3.4 | Certificate Authority |
| VikingCloud | VikingCloud | NA | Certificate Authority |
| Zero Touch PKI | ZTPKI CA | | Certificate Authority |

* CyberArk Labs has tested these versions. Other versions might be compatible but are not supported by CyberArk.

** For Dell iDRAC and HP iLO, refer to the Adaptable Application driver reference samples. See "Adaptable Application " on page 274.

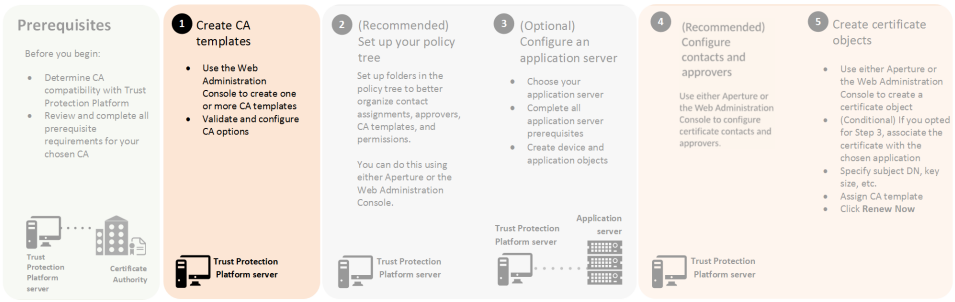
*** DigiCert PKI Platform is provided as a supported reference sample for the Adaptable CA driver. See "Adaptable CA" on page 61.

### About CSR-supported formats

Trust Protection Foundation supports the PKCS#10 CSR format. There are only a few CAs which support the CMC format, which is typically used for the special case of key archival. The CMC format provides a mechanism for the request to encrypt and pass a private key as part of a certificate's signing request. PKCS#10 CSR does not have that capability. If you need CMC support, please contact CyberArk customer support.

## Step 1: Create CA templates

During certificate enrollment and provisioning procedures, every certificate object must reference a CA template object.




CA template objects provide the information Trust Protection Foundation needs to submit the certificate signing request (CSR) to the CA and retrieve the signed certificate. The CA template also defines the type of certificate that is requested from the CA when the CSR is submitted.

**EXAMPLE** If Amazon were your CA and you wanted to automate enrollment and provisioning of your certificates using Amazon Amazon Certificate Manager, you would use Policy Tree to create a certificate template object and provide the required settings:

Using information you gathered as part of the prerequisites, you create the CA template object for the Amazon Amazon Certificate Manager using Policy Tree. You refer to the related CyberArk documentation (the Trust Protection Foundation Help system), or in the *Certificate Authority and Hosting Platform Integration Guide* related to your chosen CA. The documentation includes the information required by your chosen CA.

**NOTE** Before you attempt to create CA template, device, or application objects, you must enable the *create* permission under the folder where you want to create the new object. For more information, see Permissions Overview in the *Administration Guide*.

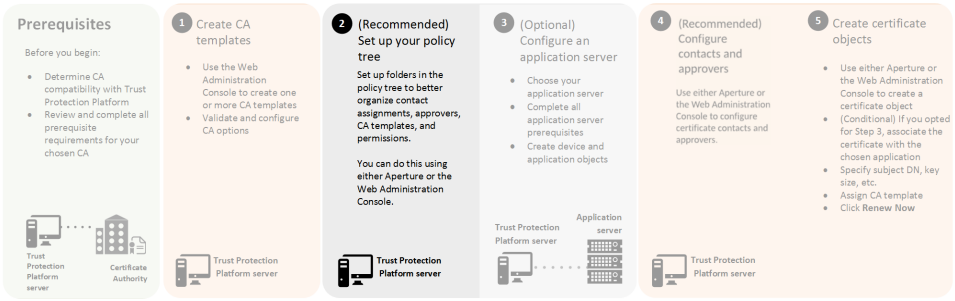
### To create a CA Template object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. From the **Tree** drop-down menu, click **Policy**.
3. In the **Policy** tree, select the folder where you want to create the CA Template object, and then click **Add**.
4. Click **CA Template**, then select a driver to create it.
5. In the **CA Name** box, type a name for the new CA driver object.
6. When finished, click **Apply**.

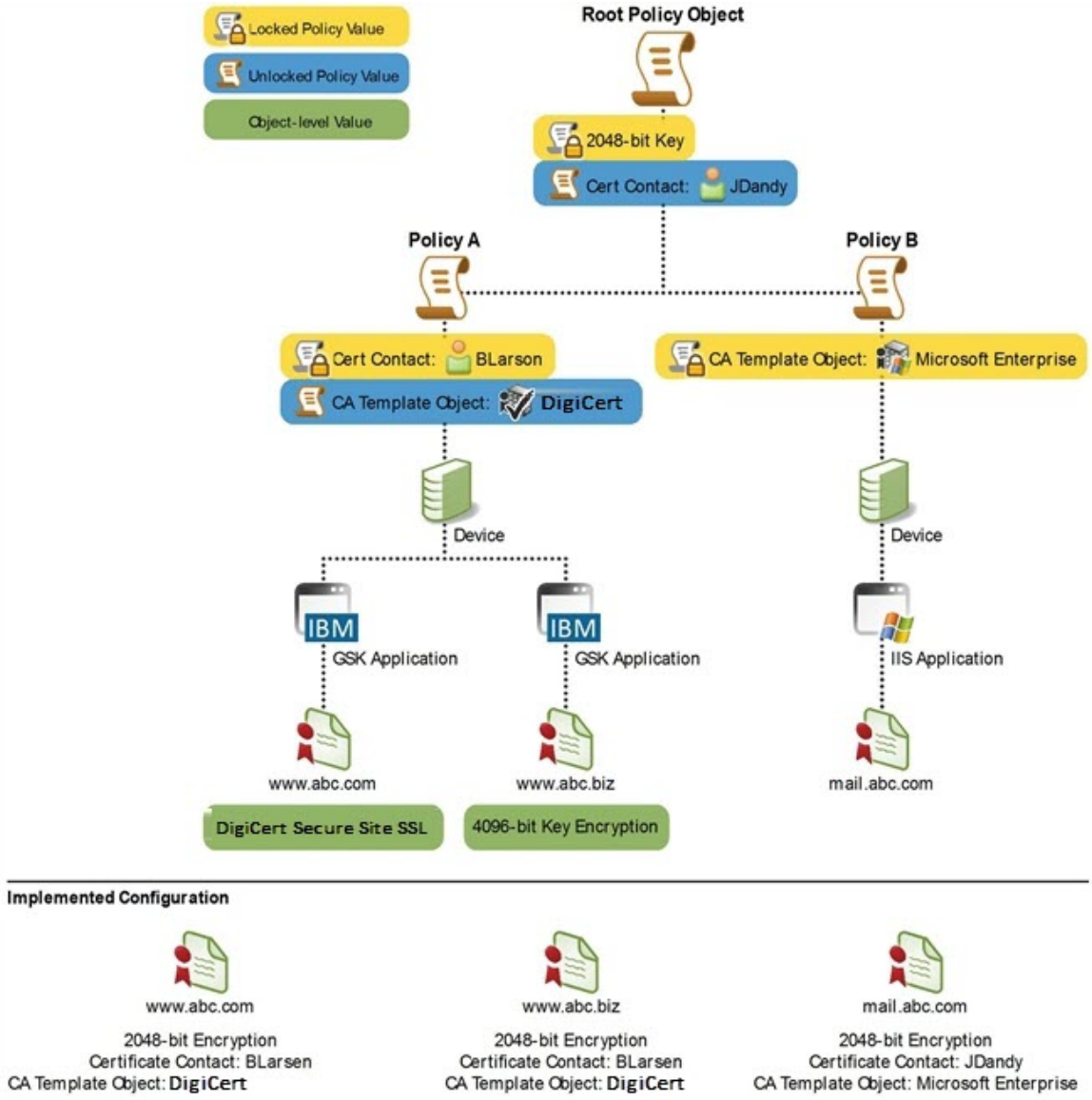
The CA Template object settings vary depending on the associated CA requirements. Refer to the CA template object settings for the CA you are configuring.

## Step 2: (Recommended) Set up your policy tree

Folders provide a hierarchical framework for applying policy settings, managing configuration and organizing objects within your environment, much like hierarchical directories such as Active Directory.




Folders can contain objects such as devices, applications, certificates, CA templates, and even other folders.

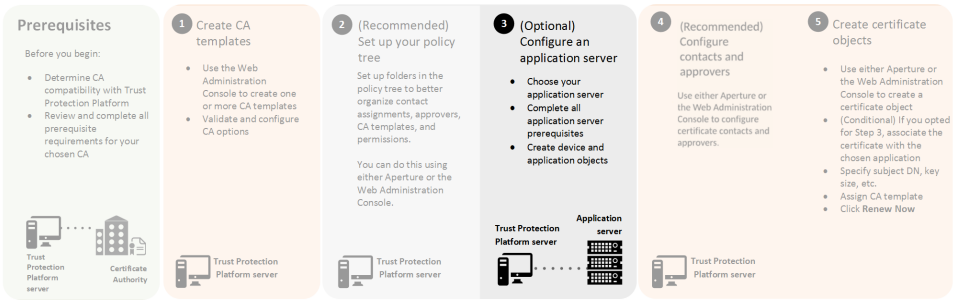



When managing CA templates, each folder enables you to set policy that applies for each object type contained below it. For example, if several certificates are contained within a folder, one or more of the policy attributes for those certificate objects can be defined on the folder. Consequently, folders can be used to standardize object configuration parameters and enforce security requirements throughout your encryption environment.

To learn more about how policy works, see Using Policies to Manage Encryption Assets in the *CyberArk Trust Protection Foundation Administration Guide*.

# Step 3: (Optional) Configure an application server

If you want to automate certificate provisioning, you can have Trust Protection Foundation install certificates for use by an application server, such as F5's BIG-IP Local Traffic Manager (F5 LTM Advanced) or Amazon Web Services (AWS). Once you've selected and configured your application server software to use a certificate, you must then create associated objects for them on the Trust Protection Foundation server (by creating device and application objects and then associating them with the relevant certificates used by your application server software).




Application objects provide the configuration information Trust Protection Foundation needs to install and validate certificates on the Application object's associated platform or keystore.

To enable Trust Protection Foundation to provision certificates to supported platforms or keystores, you must complete the following:

1. Prepare the target system.

   This includes setting up SSL on the target system and granting the required system permissions. For a listing of supported applications and their associated appendix, see "CyberArk CA and application drivers library" on page 31.

2. (Conditional) If Trust Protection Foundation interacts with a device using an SSH command line and must use a jump server to access the device because it is behind a firewall, create a Jump Server object and configure its associated Device objects.

   For more information, see "Managing Jump Server Objects" on page 226.

3. In the Policy Tree, create a Device object for the network-accessible computer system where the certificate is installed.

   For more information, see "Managing device objects" on page 238.

4. In the Policy Tree, create and configure an Application object for the network appliance or keystore where the certificate is installed.

   For more information, see "Managing application objects" on page 259.

5. In the Policy Tree, associate the Application object with the appropriate certificates.

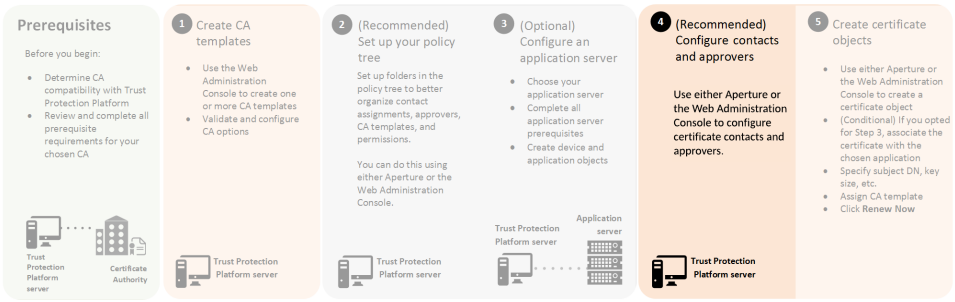
   In the case of a network appliance, you associate each Application object with the relevant certificate consumed by the network appliance. For a keystore, you associate each Application object with the certificate stored in the keystore.

   > **TIP** Application objects can only be associated with a single certificate. However, a single certificate can be associated with many application objects.

   For more information, see Associating Certificates and Applications.

## Step 4: (Recommended) Configure contacts and approvers

Who you assign as contacts and approvers of your certificates is an important part of your PKI strategy. This is especially true because employees continue to pose the greatest threat to securing trust. Typically, this is because many employees fail to follow security best practices.




> **BEST PRACTICE** When a person, who is a contact (owner) for a certificate, leaves your organization, you can easily lose track of the certificate. The lack of a contact creates a security risk. To help ensure that at least one contact receives notifications pertaining to a certificate, you should assign *groups* as contacts. By assigning groups (rather than individuals) as the contact for a certificate, you reduce the risk of un-tracked (and potentially expired) certificates.

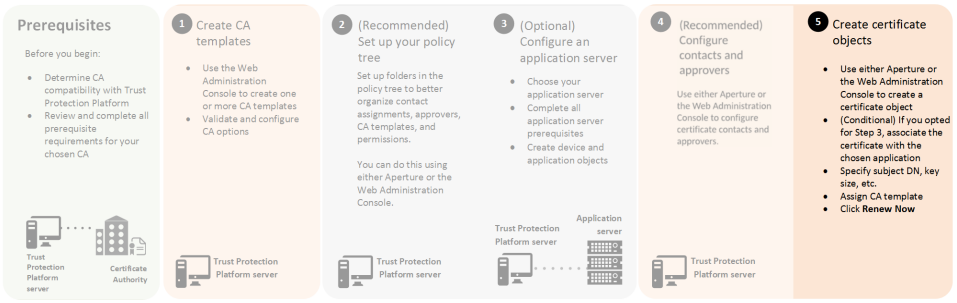
To configure contacts and approvers:

1. From the Platform menu bar, click Policy Tree.
2. In the Logging tree, be sure that SMTP channel is enabled. For more information, see SMTP channel object configuration in the *Log Server Administration Guide*.
3. To create the notification, copy an Email template and configure the email text. For example, use a **Email to Owner Certificate Expiring** template to notify contacts (Owners) about expiring certificates.
4. In Policy Tree or Aperture, policy, add at least one group or individual contact and one group or individual approver.

   By default, the contact and approver is the Trust Protection Foundation administrator. All objects under a given policy folder can inherit the contacts and approvers set on that policy folder; or you can specify contacts and approvers directly on objects within a policy folder. For more about how policy folders work, see Using policies to manage encryption assetsin the*Trust Protection FoundationAdministration Guide*.
5. (Optional) You should set up workflows to require approvers to take a manual action, such as authorizing a certificate renewal in order for the renewal to take place.

## Step 5: Create certificate objects

Certificate objects provide Trust Protection Foundation with the information it needs to monitor certificates. But how and when are they created?




When certificates are either discovered or imported, Trust Protection Foundation creates an associated certificate object automatically. However, if you create a new certificate manually, you must create a certificate object, enter the Subject DN and any SANs, and then select a CA template.

> **DID YOU KNOW?** During discovery, *device* and *application* objects are also created automatically.

For information about certificate discovery, see Discovering certificates and keys.

For more information about downloading or importing certificates, see Downloading certificates, private keys, and root chains in the *Trust Protection Foundation Certificate Management Guide*, and Discovering certificates and keys in the *Trust Protection Foundation Administration Guide*.

For more information about managing, installing, creating, and editing certificates, see Managing certificates and private keys–overview in the *Trust Protection Foundation Certificate Management Guide*.

Certificates are created in the following ways:

- **CA Import** (currently supported for Microsoft CA) - A certificate authority import job is created and Trust Protection Foundation discovers the certificates from the certificate authority and places them according to the placement rules.

  For more information, see Importing certificates from a certificate authority.

- **Instant Discovery** - Instantly scans a destination host and if certificate is discovered you can choose to manage it, which automatically creates the certificate object in the chosen destination. Instant Discovery is similar to Network Discovery, except that it scans only a single host.

  For more information see Using Instant Discovery.

- **Manually** - Manually import an existing certificate or certificate-key pair into the system using Trust Protection Foundation. If there is an application (or device) associated with the certificate, you will also need to manually create the application (or device).

  For more information, see Creating a certificate installation in Aperture, and Importing an existing certificate.

- **Network Discovery** - Scans a defined network range and, according to the placement rules set in Aperture, automatically creates and places the discovered certificates in their corresponding folders (according to the rules).

  For more information, see Discovering certificates and keys.

- **Onboard Discovery** - Automates the process of importing certificates into Trust Protection Foundation from network devices where you can then monitor, validate, and provision them.

  For more information, see Using Onboard Discovery.

- **CyberArk Server Agent** - installs the agent on local systems to performs scheduled work, including discovery of certificates and keys found in designated keystores and directories.

  For more information, see Server Agent–Introduction in the *Server Agent Installation and Configuration Guide*.

- **Web SDK** - Allows you to use REST API calls for Certificate object creation. You can use Certificates/Import to add one Certificate object or Discovery/Import to find and add an entire set.

  For more information, see POST Certificates/Import and POST Discovery/Import in the *Developer's Guide*.

# 2

# CyberArk CA and application drivers library

The CyberArk Drivers Library contains drivers designed to connect Trust Protection Foundation with some of the most common certificate authorities.

This section shows certificate authorities and provisioning target platforms that are supported by CyberArk Trust Protection Foundation. Where applicable, supported versions of the Trust Protection Foundation and integration platforms are indicated as well.

Older and newer versions might be compatible depending upon the extent to which the platform vendor or service provider have modified their application interface.

This chapter contains the following topics:

# Trust Protection Foundation enrollment and revocation driver support

This topic contains reference information showing internal and external CAs and Trust Protection Foundation versions and the host platforms on which they are supported.

The following internal CA drivers are supported on the past four versions of Trust Protection Foundation:

- Adaptable CA

  Includes a reference sample of DigiCert PKI Platform (Magnum).

- Microsoft Standalone CA
- Microsoft Enterprise CA (ADCS)
- OpenSSL
- OpenTrust Enterprise PKI
- RedHat Certificate System
- DigiCert PKI Platform
- Verizon CyberTrust UniCERT

> **TIP** For a complete list of supported version numbers, see "Supported integrations with Trust Protection Foundation" on page 18 in the *Certificate Authority and Hosting Platform Integration Guide*.

The following external CA drivers are supported on the past four versions of Trust Protection Foundation and all include an API interface and are cloud-based hosting platforms:

- Amazon Certificate Manager (ACM)
- Sectigo Certificate Manager (SCM)
- DigiCert CertCentral
- GeoTrust Reseller
- GeoTrust TrueFlex (GESC)

- GlobalSign MSSL
- HID PKIaaS
- QuoVadis
- Symantec Managed PKI for SSL
- Thawte
- VikingCloud

**IMPORTANT** In Trust Protection Foundation 26.1 and later, Entrust Certificate Services is no longer supported. Revocation of certificates originally issued by Entrust is handled by the Sectigo Certificate Manager driver.

**TIP** For a complete list of supported version numbers, see "Supported integrations with Trust Protection Foundation" on page 18 in the *Certificate Authority and Hosting Platform Integration Guide*.

## Supported certificate products for external CAs

The following table shows which certificate products are supported by which external certificate authorities (CAs).

| External CAs | Supported Certificate Products | Supported Code Signing Certificate Products |
|---|---|---|
| **Sectigo Certificate Manager (SCM)** | ■ EliteSSL<br>■ EV Multi-Domain SSL<br>■ EV SSL<br>■ Multi-Domain Instant SSL<br>■ PremiumSSL<br>■ PremiumSSL Wildcard<br>■ Unified Communications (UCC) | |
| **DigiCert CertCentral** | ■ EV Multi-Domain | ■ Code Signing |

| External CAs | Supported Certificate Products | Supported Code Signing Certificate Products |
|---|---|---|
| | ■ EV SSL<br>■ Multi-Domain SSL<br>■ Premium<br>■ Private Premium<br>■ Private Multi-Domain SSL<br>■ Private SSL<br>■ Private SSL Wildcard<br>■ Standard SSL<br>■ Wildcard | ■ EV Code Signing |
| **Entrust Certificate Services (Entrust)** | ■ Advantage<br>■ EV Multi-Domain<br>■ Payment Services Directive (PSD2)<br>■ Private Dedicated SSL<br>■ Private SSL<br>■ Qualified Web Authentication Certificate (QWAC)<br>■ Standard<br>■ UC Multi-Domain<br>■ Wildcard | ■ Code Signing<br>■ EV Code Signing |
| **GeoTrust TrueFlex (GESC)** | ■ Enterprise SSL EV | |

| External CAs | Supported Certificate Products | Supported Code Signing Certificate Products |
|---|---|---|
| | ■ Enterprise SSL | |
| **GeoTrust Reseller** | ■ Secure Site<br>■ Secure Site Pro<br>■ Secure Site with EV<br>■ Secure Site Pro with EV<br>■ True BusinessID<br>■ True BusinessID with EV | |
| **GlobalSign MSSL** | ■ Extended Validation (EV) SSL<br>■ Organization SSL | |

| External CAs | Supported Certificate Products | Supported Code Signing Certificate Products |
|---|---|---|
| **Symantec Managed PKI for SSL** | ■ OFX SSL<br>■ Private SSL<br>■ Premium EV SSL<br>■ Premium SSL<br>■ Premium Intranet SSL<br>■ RapidSSL Enterprise<br>■ Standard SSL<br>■ Standard EV SSL<br>■ Standard Intranet SSL | |
| **Thawte** | ■ SSL Web Server<br>■ SSL Web Server EV | |
| **VikingCloud** | ■ Domain Validated SSL<br>■ Enterprise SSL<br>■ EV SSL<br>■ Internal SSL<br>■ Premium SSL<br>■ SAN SSL<br>■ TrustKeeper SSL Plus<br>■ TrustKeeper Plus EV SSL<br>■ Wildcard SSL | |

## How long does it take for a certificate authority (CA) to issue a certificate?

Some certificate authorities can issue certificates almost instantly while others can take up to an hour or more to act on a CSR enrollment.

To prevent unnecessary resource utilization, and to avoid overloading the CA with repeated duplicate requests, Trust Protection Foundation employs a system that periodically checks to see if the certificate has been issued. If the certificate has not been issued, the duration between requests gradually increases until it reaches 32 minutes.

By default, the retry interval period is 30 seconds, then 1 minute, 2 minutes, 4 minutes, 8 minutes, 16 minutes, and finally 32 minutes.

Once the duration reaches 32 minutes, the system will continue to check every 32 minutes until the certificate is issued. Therefore, the longer it takes the CA to approve the request, the longer it will take to import a certificate. For example, if it takes the CA a day to complete the request, it will take Trust Protection Foundation up to 32 minutes after the certificate is issued to be imported into the system.

> **NOTE** Some TPP drivers have a different retry interval period because the CA may return a certificate before the default 30 seconds. These CA templates and their intervals are:
>
> Adaptable CA, Amazon CA (using private CA), and Google Private CA - 1 second, 2 seconds, 10 seconds, 30 seconds, 1 minute, 2 minutes, 4 minutes, 8 minutes, 16 minutes, then 32 minutes
>
> Amazon CA (using public CA) - 10 minutes
>
> VikingCloud - 30 minutes

You may wonder what are the primary causes of delays in certificate issuance?

There are several reasons why certificate issuance can be delayed:

1. The certificate authority is just slow. Some certificate authorities regularly take 30 minutes to an hour to act on a CSR.
2. Sometimes organizations have configured a CA approval for which a manual approval is required before a certificate can be approved. In those cases, the delay will be as long as it takes for the approver to receive and act on the request for approval.

> **NOTE** This is completely outside the control of Trust Protection Foundation. This is a setting controlled by the CA when approvals are required. If your organization is one that uses this type of setup, we encourage you to look at settings in Trust Protection Foundation that perform the same function. These include Certificate Workflows and the Adaptable Workflow feature which can integrate with external approval systems, like, for example, ServiceNow where approvals can be managed.
>
> For more information on certificate workflows, see Implementing Certificate workflow management.
>
> For more information on Adaptable Workflow, see About Adaptable Workflows.

3. Sometimes the certificate authority needs the customer to re-verify ownership of the domain before it will issue a certificate for the requested domain.

If Trust Protection Foundation seems slow importing the certificate, the delay is likely on the certificate authority's side. Trust Protection Foundation will continue checking every thirty minutes or so until the certificate is issued and can be imported into the system. If you have questions, check with your system administrator, or the person who administers manual approvals for your certificate authority.

# Certificate installation (provisioning) driver support

When referencing the table below, keep the following requirements in mind:

- On Windows platforms where SSH is required for connection, copSSH v1.4.3 or Tectia Server v6 is required.
- On systems where OpenSSH is used, OpenSSH version 4.4 or later is recommended for correct operation. OpenSSH 4.3 is supported (with some restrictions).
- Jump Server support for SSH-based application drivers is included with Trust Protection Foundation 6.4 and later and requires OpenSSH v4.4, Tectia 4.4.12, or Tectia 6.0.9.24 on the jump server.
- For updated information related to iControl, see "iControl considerations and background" on page 412.

## Common keystores

The following provisioning drivers are supported on the past five versions of Trust Protection Foundation, except where noted:

- Adaptable Application
- Apache/PEM (OpenSSL)
- IBM Global Security Kit (GSK)
- Imperva MX
- Java Keystore (JKS and JCEKS)
- Microsoft CAPI
- Microsoft IIS
- Network Security Services (Oracle iPlanet)
- PKCS#12

  See "About CSR-supported formats" on page 22 in the Certificate Authority and Hosting Platform Integration Guide for more information.

**TIP** For a complete list of supported version numbers, see "Supported integrations with Trust Protection Foundation" on page 18 in the *Certificate Authority and Hosting Platform Integration Guide*.

**NOTE** For a list of supported keystores to which you can provision using the Server Agent, see Server Agent supported keystores in the *Certificate Authority and Hosting Platform Integration Guide*.

## Supported appliances

The following appliance drivers are supported on Trust Protection Foundation:

- Amazon Web Services IAM/ELB/CloudFront
- Apache
- Azure Key Vault (Microsoft)
- CAPI (IIS 7+)
- Citrix NetScaler MPX (with HSM)
- Citrix NetScaler VPX

- F5 Big-IP F5 LTM Advanced
- Google Cloud Load Balancer (external proxies only)
- HashiCorp Vault PKI
- IBM GSK
- IBM Sterling Connect:Direct
- IBM WebSphere DataPower
- iPlanet
- JKS
- Juniper SA/MAG Series
- Palo Alto Networks Next Generation Firewall
- PEM
- Riverbed SteelHead WAN Optimizer
- Tealeaf PCA (Passive Capture Appliance)
- VAM nShield

**TIP** For a complete list of supported version numbers, see "Supported integrations with Trust Protection Foundation" on page 18 in the *Certificate Authority and Hosting Platform Integration Guide*.

## Supported integrations: devices, applications, services and features supported by CyberArk

**PKIX-Aligned Policy Enforcement:** Supported drivers strictly enforce standard certificate profiles and centrally defined cryptographic policies. If a target application rejects the requested algorithm, the driver returns an explicit error instead of silently downgrading the key. Refer to the following table for drivers supporting PKIX-alignment.

To see which devices, applications (and related features), and services are supported by CyberArk Trust Protection Foundation, review the HTML version of this topic online at https://docs.venafi.com/Docs/current/TopNav/Content/Drivers/r-drivers-applicationDrivers-supportedFeatures.htm.

Once you're in the online documentation, search for "Supported integrations: devices".

To see our online help, visit https://docs.venafi.com. We no longer require a Support account to view our public documentation site.

## Supported shells

Regardless of the command-line interpreter (shell) that you use, Trust Protection Foundation switches to a Bourne shell. If the default shell for the service account is something other than Bourne (/bin/sh), the agentless shell commands might not work properly.

> **NOTE** CyberArk recommends that the Bourne (sh) shell is set as the default shell for the Service Account.

## About keystore backups

Whenever Trust Protection Foundation performs a certificate or private key operation, it creates a backup copy of the crypto file in the same directory where the original keystore resides.

If a backup operation fails, Trust Protection Foundation logs an event and halts processing. For example, if a backup operation fails for the JKS driver, the *JKS - Backup KeyStore Failed* event (ID 400E0020) is logged.

> **CAUTION** If a CyberArk driver cannot back up a crypto file, it will not attempt any further certificate and private key operations.

## About certificate lifecycle management

Trust Protection Foundation simplifies the process of managing digital certificates throughout their lifecycle. When a certificate is brought under management, Trust Protection Foundation monitors the certificate and provides current information about the certificate status. When a certificate nears the end of its lifecycle, Trust Protection Foundation provides notifications so you can renew and install the certificate before it expires.

If a network certificate is configured for Enrollment, Trust Protection Foundation interfaces directly with the CA to initiate certificate renewal and key generation requests according to organization-defined workflow and approved folders. After the CA signs the certificate, Trust Protection Foundation retrieves the certificate and securely stores it in the Secret Store. The administrator can then download the certificate from the Secret Store and install it on the target system(s).

If a network certificate is configured for Provisioning, Trust Protection Foundation automatically requests, renews, and installs the certificate on its associated application(s), ensuring that the certificate is reliably deployed and managed.

> **NOTE** Individual stages may vary per application. For information on the certificate lifecycle stages for each application, see "Protecting server platforms and keystores" on page 223 in the *Certificate Authority and Hosting Platform Integration Guide*.

The following table outlines the managed stages of the network certificate lifecycle.

Certificate enrollment involves two distinct "todo" items. The Certificate ToDo (stages 0-450), and Certificate Enrollment ToDo (stages 500-750). These ToDos respect different engine assignments. For example, the Certificate ToDo respects what is specified for the certificate object, but the Certificate Enrollment ToDo respects what is specified on the CA Template. If you are seeing certificate processing failures (including while using Intune), try setting the engine assignment to be the same on both the certificate object and the CA template.

> **NOTE** If the private key and CSR are locally generated on the Trust Protection Foundation server, stages 0-700 are performed by the default X509 Certificate Application driver. Stages 0-700 are only performed by the certificate's consumer Application driver if the private key and CSR are remotely generated on the certificate's consumer application.
>
> The private key and CSR are remotely generated on the certificate's consumer applications if the **Generate Key/CSR on Application** option is enabled in the Certificate object.

# Stage Codes and Descriptions

## Certificate Stage Codes

| Stage Code | Friendly Name | Description |
|---|---|---|
| 0 | StartProcessing | Prepares the certificate for lifecycle processing. |
| 100 | CheckStore | Applies only to remote generations.<br><br>If the private key and CSR are generated remotely, Trust Protection Foundation compares the keystore or Directory configuration parameters specified in the Application object with the actual configuration on the application. |
| 200 | CreateConfigureStore | Applies only to remote generations.<br><br>If the certificate keystore does not exist, Trust Protection Foundation creates the keystore as per the configuration parameters defined in the Application object. |
| 300 | CreateKey | Creates the private key.<br><br>**DID YOU KNOW?** Stage 300 is used for key generation only when the API of a target device separates keypair and CSR generation. When they're combined, both key and CSR generation are always done at stage 400. |
| 400 | CreateCSR | Creates the Certificate Signing Request (CSR). If Service Generated CSR is enabled and the certificate is associated with multiple applications, the CSR will be centrally generated so Trust Protection Foundation can push the private key to multiple applications. |
| 500 | PostCSR | Submits the CSR to the Certificate Authority (CA).<br><br>If you post a manual CSR, this is the first stage of the certificate lifecycle. |
| 600 | ApproveRequest | Approves the certificate renewal at the CA. |
| 700 | RetrieveCertificate | Retrieves the certificate from the CA. |
| 800 | InstallCertificate | Installs the certificate on the target application. This provisioning happens in several stages, such as 801, 802, etc. All stages between 800-899 are provisioning stages. |

| Stage Code | Friendly Name | Description |
|---|---|---|
| 900 | CheckConfiguration | Reserved for future use. You cannot apply this stage to a workflow. |
| 1000 | ConfigureApplication | Reserved for future use. You cannot apply a stage to this workflow. |
| 1100 | RestartApplication | Used for command injection workflow that must be executed after a certificate has been successfully provisioned. |
| 1200 | EndProcessing | Completes the certificate processing and, if configured, runs a Validation check on the certificate and application. |
| 1400 | Revocation | Submits a revocation request to the CA.<br><br>Certificate revocation is a certificate operation; it does not involve the application driver. |
| 1500 | UpdateTrustStore | Updates the Trust Store at the host to comply with the effective bundle. To learn more, see Viewing the Effective Bundle. |
| 1600 | EndTrustStoreProcessing | Completes the processing of the Trust Store. |

# 3

# CA integration setup

The CA Template object defines the information Trust Protection Foundation needs to connect to a certificate authority (CA) so that it can submit a certificate signing request (CSR) and retrieve certificates from the CA during enrollment and installation operations. The CA Template object also defines the template used to post the certificate CSR. This section provides the information you need to create and manage CA Template objects using Trust Protection Foundation.

CyberArk Trust Protection Foundation™ can submit CSRs and retrieve certificates from the following CAs:

- Amazon Certificate Manager (ACM)
- Sectigo Certificate Manager (SCM)
- DigiCert
- GeoTrust Reseller
- GeoTrust TrueFlex
- GlobalSign MSSL
- Microsoft Certificate Services–Enterprise and Standalone editions
- OpenSSL
- OpenTrust Enterprise
- QuoVadis
- RedHat Certificate System
- RSA Keon
- Google Cloud CA Service
- Self-signed
- Symantec Managed PKI for SSL (Symantec MPKI)
- Thawte
- VikingCloud
- Verizon UniCERT

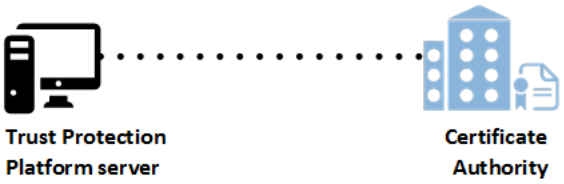
This chapter contains the following topics:

# Certificate Authority template overview

This section provides an overview of how Certificate Authority (CA) template objects are used in CyberArk Trust Protection Foundation™ to manage the certificate lifecycle.

## CA template objects and lifecycle management

CA Template objects are required at the Enrollment and Provisioning levels of certificate management. They provide Trust Protection Foundation with the information it needs to connect to the CA.




This connection enables Trust Protection Foundation to submit certificate signing requests (CSRs) and retrieve certificates from the CA during enrollment and provisioning operations. The CA Template object also defines the template used to post the certificate CSR.

Under certificate enrollment and provisioning, every Certificate object must reference a CA template object. Otherwise, Trust Protection Foundation cannot request the certificate or renew an existing certificate.

## CA template object hierarchy

CA Template objects are created and managed in the Policy tree. To access the Policy tree, select **Policy** from the Tree.

The Policy tree provides a hierarchical view of your encryption deployment model, so CA Template objects appear in context of other system objects such as Policy, Application, and Certificate objects.

In the Policy tree hierarchy, CA Template objects are created only under folder.

## Managing multiple instances of the CA configuration

Trust Protection Foundation allows you to create multiple CA Template objects for each CA, if needed, to support multiple instances, accounts, or configurations of the CA. For example, in the case of Microsoft Certificate Services, you might have separate instances of the CA installed on multiple servers. In this case, you would create a separate CA Template object to support the individual CA requirements.

You can also create multiple CA Template objects for each CA to support different certificate templates. For example, if your CA offers multiple types of certificates, such as *organization validation* (OV) and *extended validation* (EV), you must create two different CA Template objects—one that issues OV SSL certificates and one that issues EV SSL certificates.

## Understanding CA overrides and compliance

Understanding CA Overrides and Compliance When you configure a Certificate Authority (CA) template in Trust Protection Foundation, you are defining the pathway and rules for how the system requests certificates from your CA. However, it is important to understand that the CA retains ultimate authority over the contents of the issued certificate.

Many CAs enforce strict internal policies and profiles. If a user submits a certificate request containing values that conflict with the CA's template constraints, the CA may automatically override or adjust those fields to ensure compliance before issuing the certificate.

Common examples of CA-driven overrides include:

- Validity Periods: A user requests a 2-year certificate, but the CA policy strictly limits issuance to 1 year.
- Subject DN Fields: A user inputs a custom Organization or Country, but the CA template forces specific, pre-vetted values.

### Lightweight Validation and Silent Adjustments

Trust Protection Foundation's WebSDK is designed to be highly flexible and lightweight. It intentionally does not perform strict, preemptive validation against every possible CA template constraint prior to submitting the request. This design is crucial for supporting complex environments where multiple CA templates might exist under a single folder policy.

**IMPORTANT:** Because these constraints are enforced directly by the CA during the issuance process, Trust Protection Foundation considers the successful issuance a completed task and does not generate warning logs when a requested value is overridden.

### Verifying Issued Certificate Details

To ensure that your issued certificates meet your organizational requirements, you should adopt a standard practice of reviewing the **Compliance** tab on the certificate object after issuance.

The **Compliance** tab serves as your source of truth, allowing you to quickly compare the certificate details you originally submitted against the actual details the CA included when it issued the certificate. If you notice unexpected adjustments, you may need to review the profile settings directly within your CA's administrative console or adjust your Trust Protection Foundation policies to match the CA's constraints.

## About creating CA template objects

During certificate enrollment and provisioning procedures, every certificate object must reference a CA template object. CA template objects provide the information Trust Protection Foundation needs to submit the certificate signing request (CSR) to the CA and retrieve the signed certificate. The CA Template object also can be used to create the certificate CSR.

### Creating CA template objects

For detailed configuration information about each available CA template object, refer to the relative CA template topics in this section.

You must enable the *create* permission under the policy where you want to create the CA Template object.

#### To create a CA Template object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. In the **Policy** tree, select the policy where you want to create the CA Template object, and then click **Add**.
3. Select the CA Template object you want to create.
4. In the **Create** dialog, type a name for the new CA Template object, and then click **Create**.

The Detail View displays the CA Template object's associated settings.

5. Define the CA Template object settings.

   The CA Template object settings vary depending on the associated CA requirements.

   For details about CA template object settings, refer to the help topic for the specific integration you're using. For example, if you are using Amazon Amazon Certificate Manager, refer to the topic "Integrating Amazon Certificate Manager with Trust Protection Foundation" on page 104.

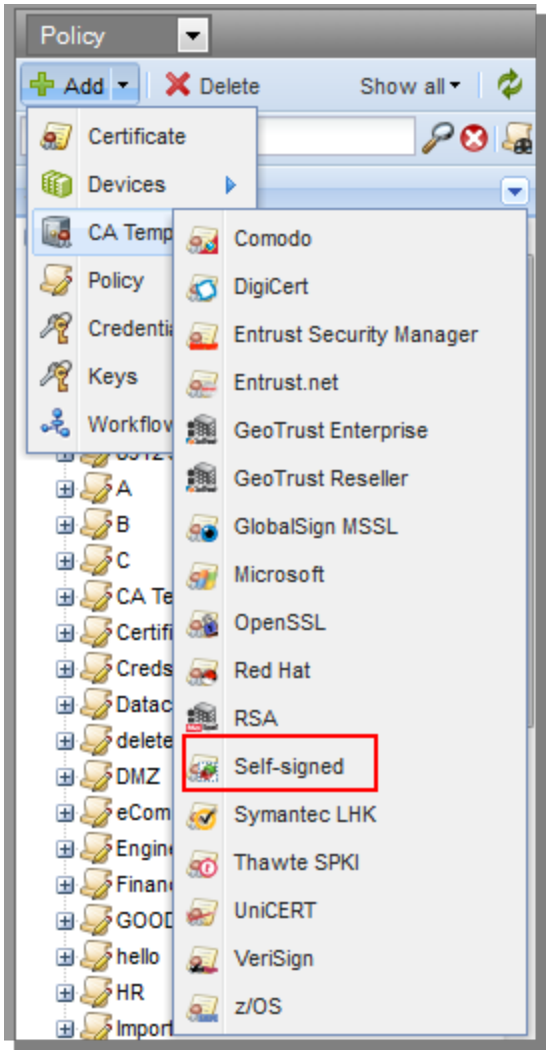
6. When finished, click **Apply**.

## Configuring a self-signed certificate template

You must have *view* and *write* permissions on the object where you want to configure settings.

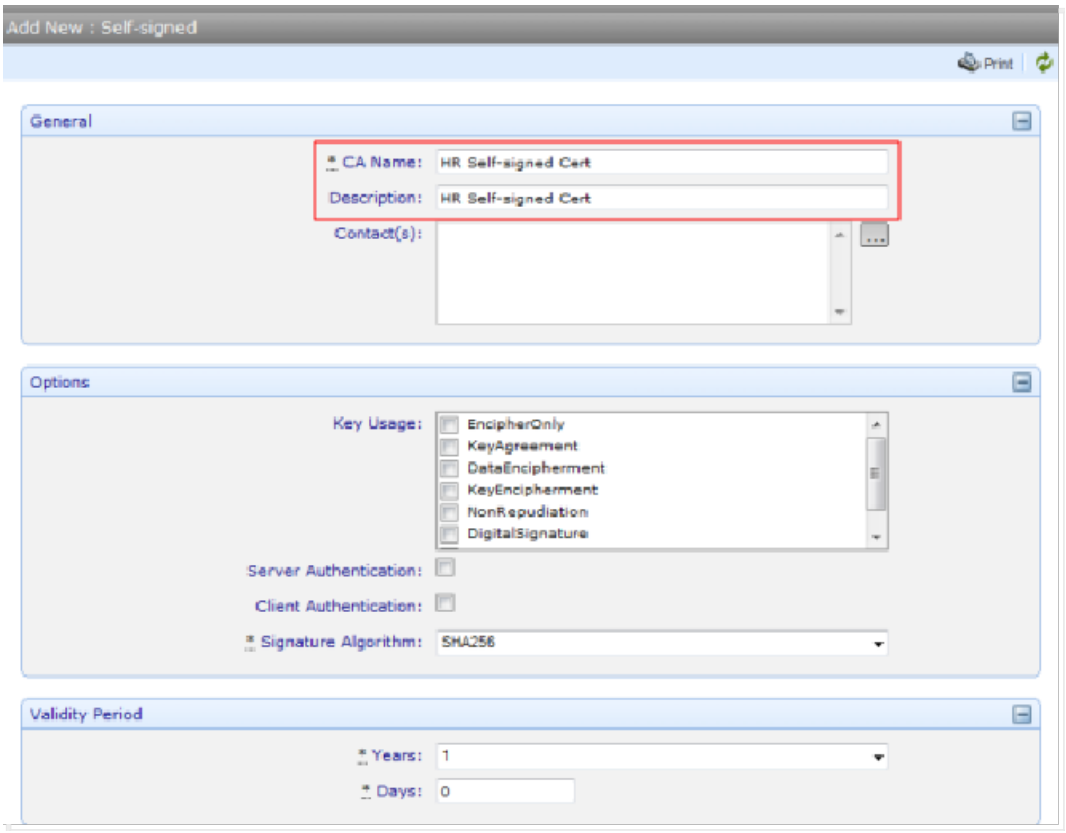
## To create a self-signed certificate template

1. From the **Policy** tree, click **Add** > **CA Template** > **Self-signed**.




2. In the **General** box, give the template a **CA Name** and **Description**.

   Specify a contact, if needed. In this example, our template is called *HR Self-signed certificate*:

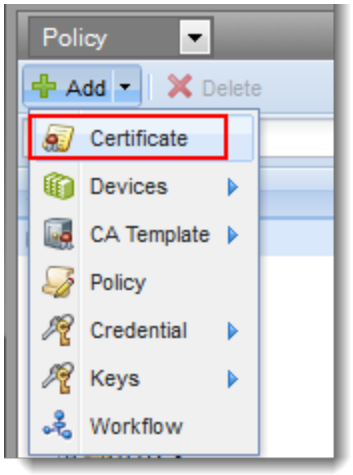



3. In the **Options** box, select:
   a. Key Usage: Select the appropriate key usage.
      i. (Optional) Server Authentication: Enhancement to Key Usage
      ii. (Optional) Client Authentication: Enhancement to Key Usage
   b. Signature Algorithm: The default is SHA256.
4. In the **Validity Period** box, specify how long the certificate will be valid.

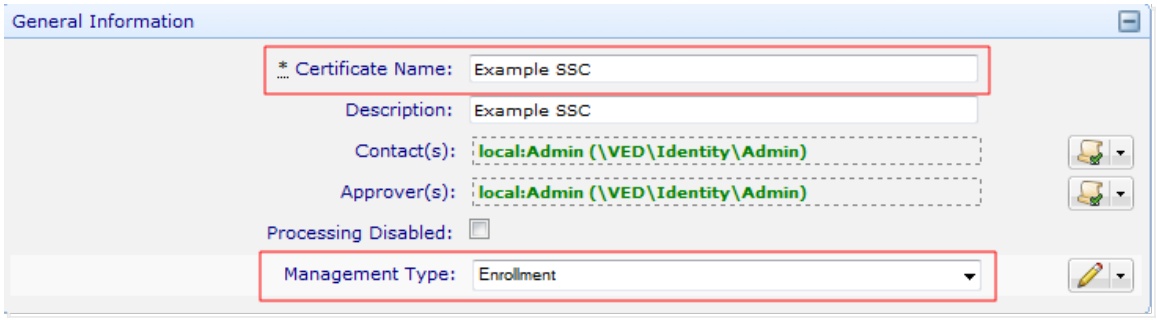
   The maximum period is 5 years.
5. Click **Save**.

### To create a self-signed certificate using the Self-signed Certificate template

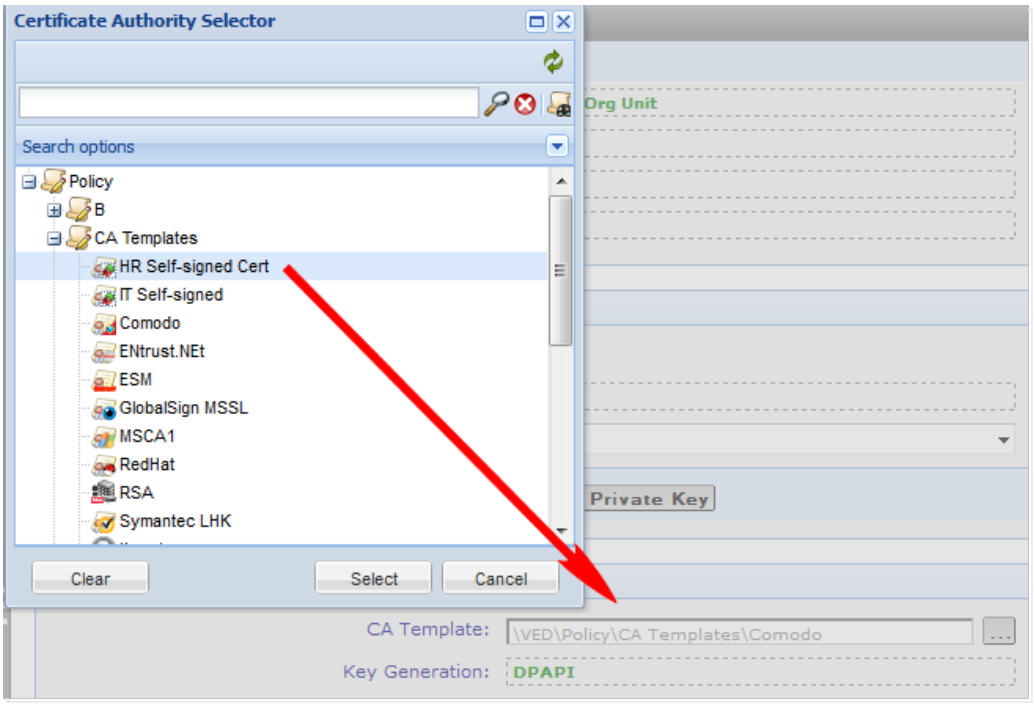
1. In Policy, click **Add** , and then click **Certificate**.




2. In the **General Information** box, fill in the **Certificate Name** and **Description**.
3. Verify that the Contact and Approver names are correct.
4. Set the **Management Type** to either Monitored, Enrollment, Provisioned, or Unassigned.

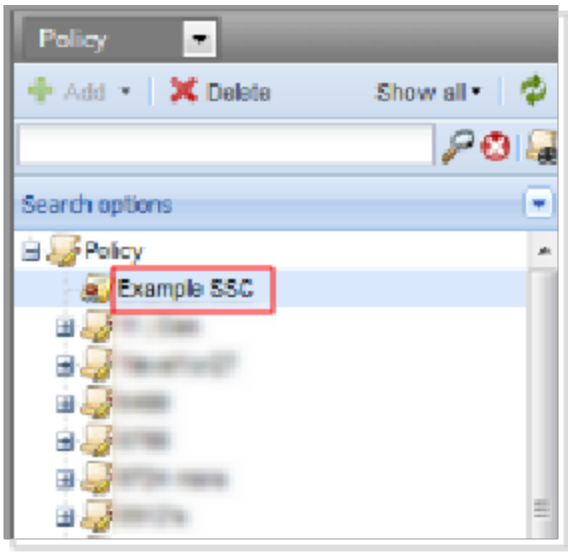



5. Fill out the certificate information.
6. In the **Other Information** box, the CA Template should be set to the self-signed certificate template you created.




7. When you're done, click **Save**.

You'll see the new certificate in the Policy tree.



## Managing CA templates using policies

In Trust Protection Foundation, a folder is a policy. Folders let administrators define global configuration parameters for assets associated with a policy, including CA template, device, and application objects.

When an object is defined under a folder (policy), that object is then associated with the folder and is subject to that folder's settings. Trust Protection Foundation reads values for the folder's subordinate objects. For example, if you configured the Device Host Information settings in a folder object, Trust Protection Foundation would read those values for the policy's subordinate device objects. Thus, policies can be used to standardize object configuration parameters and enforce security requirements.

Thus, policies can be used to standardize object configuration parameters and enforce security requirements.

For more detailed information on how folders work, see Using policies to manage encryption assets in the *CyberArk Trust Protection Foundation Administration Guide*.

> **NOTE** You must have *write* permissions on the Policy object where you want to configure the CA Template object settings.

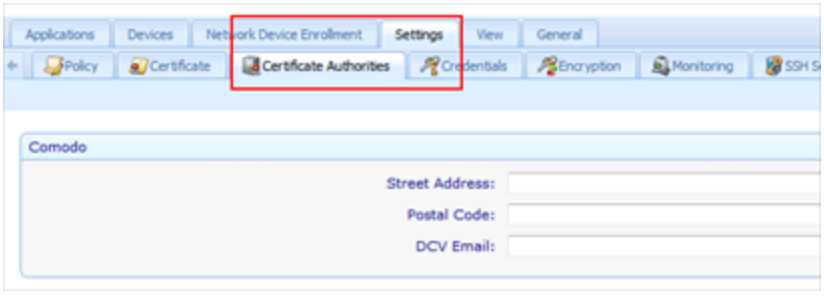
### To access the CA Template settings in the Policy object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.

   You must have *write* permission on the Policy object where you want to configure the CA Template object settings.

2. In the Policy tree, select **CA Templates**.

3. In the **Settings** tab, click **Certificate Authorities.**




   The Detail View shows all of the settings that are configurable for each CA template.

4. Find the CA template you want, then define the settings you want to apply to the policy's subordinate CA template objects.

5. Lock or unlock the value of each attribute.

   *Locked* attributes cannot be modified in objects contained in that policy, including sub-folders.
   *Unlocked* attributes are suggested values. They can be accepted or overwritten in sub-folders.

   By default, policy values are unlocked  .

For details about CA template object settings, refer to the help topic for the specific integration you're using. For example, if you are using Amazon Amazon Certificate Manager, refer to the topic "Integrating Amazon Certificate Manager with Trust Protection Foundation" on page 104.

## Understanding CA template chaining

During the certificate enrollment process, certificate chains are automatically populated from the certificate authority.

When building a chain for a certificate, TLS Protect can use three sources of information

- Chains from the CA during enrollment
- Certificates on the roots tree
- Windows certificate store

You can remove chain options that you do not want or add additional options when you need more granular control.

The **Certificate Chain Preferences** tab allows you to configure how certificates are chained for a CA Template object. This tab is visible for every CA template object, except self-signed CAs. If no issuing certificate is populated, the option defaults to automatic chaining options.

After saving the issuing certificate, you can select their preferred chain option when multiple options are available. If the selected intermediate is not the actual issuing certificate, it will be overwritten by the actual issuing certificate upon issuance.

The chain preferences assigned to a CA template object are only effective for certificates assigned the CA template object via policy or directly on the certificate object. If you have different chaining needs for

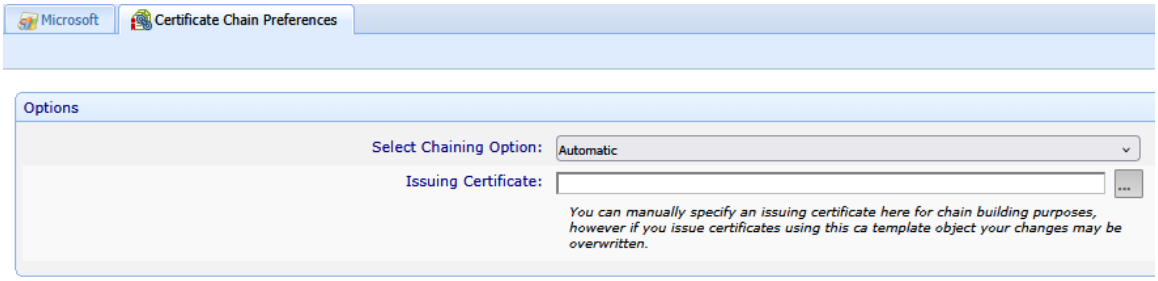
different policy folders, you can create multiple CA template objects, configure each with the desired chaining, and assign them to the appropriate policy folders. The chaining of the certificate is determined by the certificate's assigned or effective CA template object.

**NOTE** Chain options are automatically re-populated each time a certificate is enrolled. If a chain is manually removed, it will be re-populated if the certificate is re-enrolled.

**NOTE** Not all drivers support automatic certificate chain options.

### To view or add certificate chain options.

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. In the **Policy** tree, select the CA template.
3. Select the **Certificates Chain Preferences**tab.




4. Select your desired chaining option and issuing certificate.

   You can manually specify an issuing certificate for chain building purposes; however, any changes you make may be overwritten when certificates are issued using this CA template object.

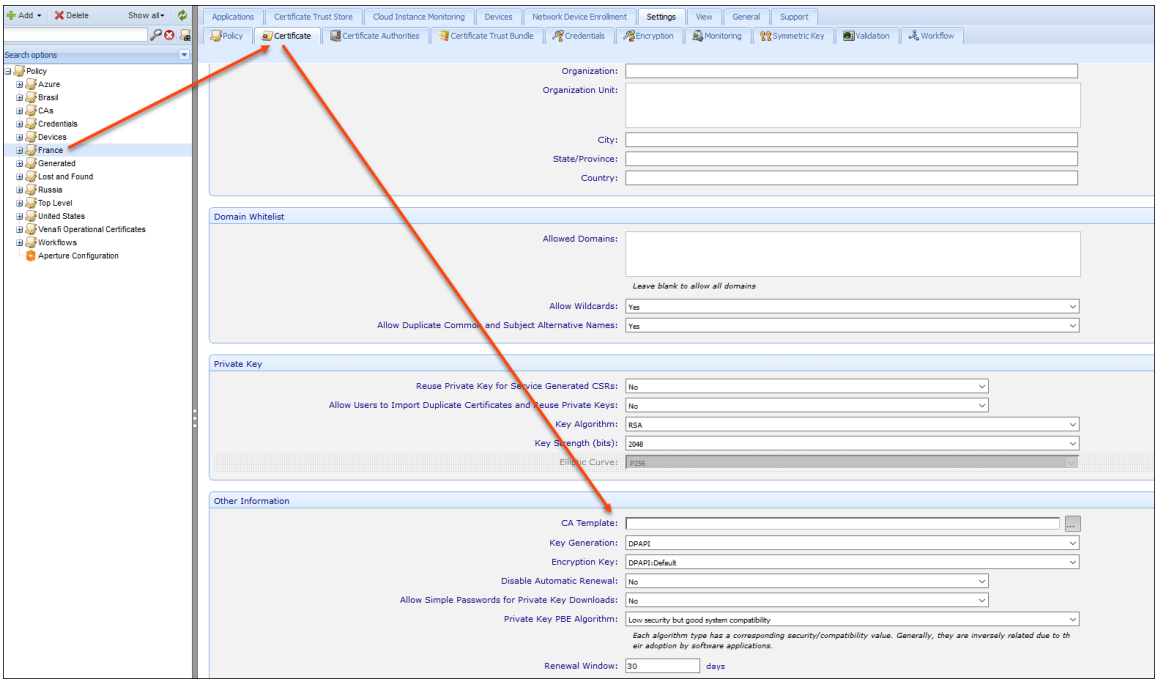
5. Click **Save**.

## Assigning a CA template to a policy folder

A CA template can be applied at the policy level. The template will affect all certificates in the folder.

### To apply a CA template to a policy

1. Click the policy to which you want to apply a CA template.
2. Click the **Certificate** tab.




3. Under **Other Information**, click  to select the appropriate CA template from the list.
4. When you're finished, click **Save**.

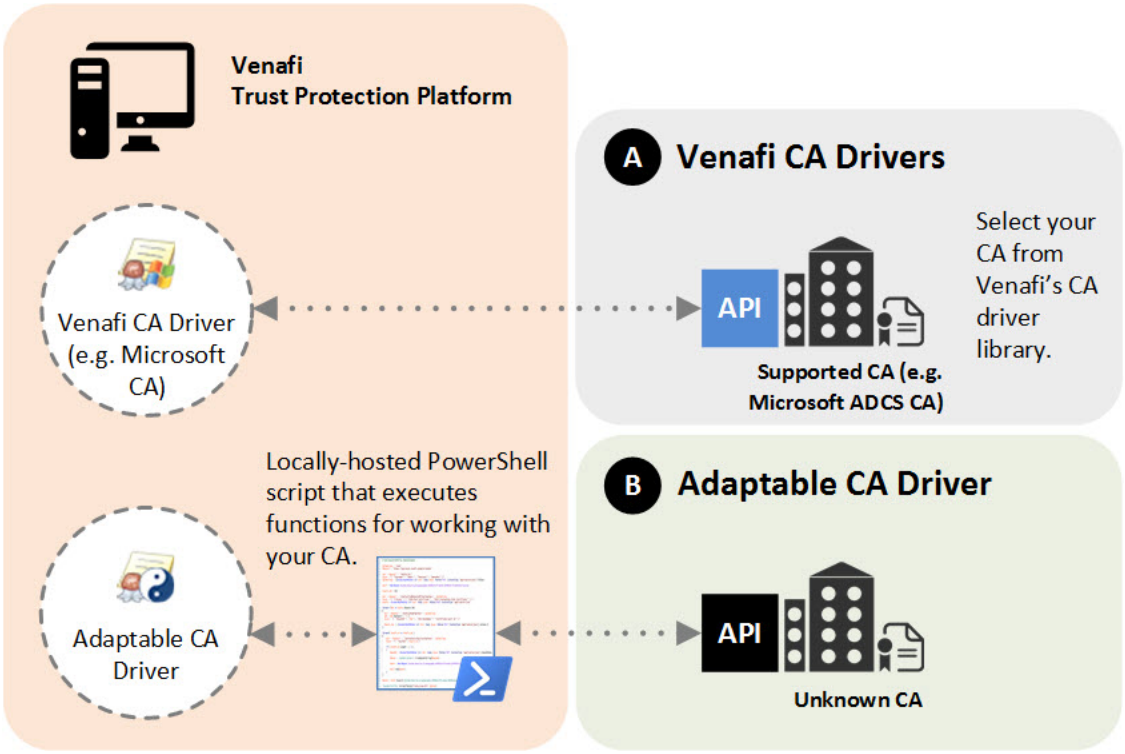
# Adaptable CA

CyberArk builds and delivers software drivers designed to connect with many of the most common certificate authorities (CAs), application servers, and enterprise monitoring systems for notification use cases. But if the CyberArk Driver Library does not include the driver you need, you can use a Venafi Adaptable Driver. In fact, because they are customizable, you can use an Adaptable driver in place of an existing driver to provide tighter integration between your business processes and Trust Protection Foundation.

Adaptable drivers provide a common set of variables required by the majority of applicable use cases that are supported by Trust Protection Foundation natively. Some Adaptable drivers also let you define additional text fields, yes/no (Boolean) fields, and a password credential field, which you can then use to elicit different behaviors or to pass additional data to the system or service to which you are integrating.

Adaptable drivers depend on a Microsoft PowerShell script hosted in your local environment to execute functions corresponding to standard certificate lifecycle stages or Trust Protection Foundation events.

**NOTE** CyberArk's Adaptable drivers can be used when a CA or application driver does not yet exist, or they can be used in place of an existing driver that requires tighter integration between your business and Trust Protection Foundation than the custom driver provides.




**NOTE** To work effectively with any CyberArk adaptable solution, you must have some working knowledge of PowerShell scripting, or you must have equivalent experience with a scripting language similar to PowerShell.

## Adaptable Certificate Authority (CA) attributes

In the **TLS Protect > Policy Tree > Certificate Authorities** section, you can configure the following Adaptable CA attributes.

- **Enable Linked Certificate Selector** allows you to configure linked certificate information when requesting a certificate. Some Certificate Authorities (CAs) can use linked or parent certificate information when issuing additional related certificates. For instance, DigiCert can issue duplicate certificates when the order or transaction ID of the parent certificate is specified. This feature enables you to specify the location of the linked object and the attribute to read for the order ID or other configuration data. The information will be included in the adaptable CA script within the specific hashtable as the linked certificate's attribute data during the SubmitCSR stages.
- **Linked Certificate** allows you to specify a certificate that will be linked with a parent or duplicate certificate. This link can be beneficial when working with Certificate Authorities (CAs) that have the capability to issue child or duplicate certificates.Linked Certificate:
- **Linked Certificate Attribute Name** allows you to specify the name of the configuration attribute that will be read from the linked certificate object. This attribute will then be passed to the PowerShell script's specific hashtable as the linked certificate's attribute data.

## What's next?

Before implementing Adaptable CA drivers, carefully review prerequisites and learn more about the required PowerShell script.

See "Adaptable CA prerequisites" below and "About the Adaptable CA PowerShell script" on the next page.

## Adaptable CA prerequisites

Before you configure the Adaptable CA template and certificate object settings in Trust Protection Foundation, you must have done the following:

- Review the Adaptable Flow design to understand where you want the custom logic to be applied.
- To work effectively with any CyberArk adaptable solution, you must have some working knowledge of PowerShell scripting, or you must have equivalent experience with a scripting language similar to PowerShell.
- Install Windows PowerShell 3.0 or later.
- Install .NET 4.8 or later.

- Make sure you have a username credential that is valid for the device you will connect to. This is the username and password SSH Manager for Machines will use to connect to the device. For information on working with credentials, see Creating user name or password credentials.
- Check out the sample script in the `\Venafi\Scripts\AdaptableSSHManagement\sample\` folder. This will help you create your script. Your scripts should be stored in the `\Venafi\Scripts\AdaptableSSHManagement\` folder. You will only be able to select items that appear in that specific folder, not any descendant folders.

## What's next?

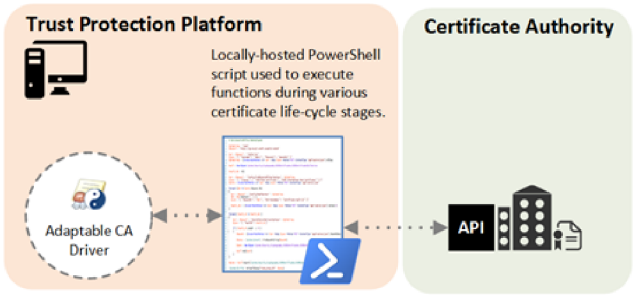
After you've completed the Adaptable CA prerequisites, see "Configuring the Adaptable CA object" on page 67.

To set up an Adaptable CA for Code Sign Manager - Self-Hosted, see Create Adaptable CA Template for Code Sign Manager - Self-Hosted.

## About the Adaptable CA PowerShell script

The Adaptable CA driver utilizes a PowerShell script that contains functions called during various stages of the certificate lifecycle.

> **NOTE** To work effectively with any CyberArk adaptable solution, you must have some working knowledge of PowerShell scripting, or you must have equivalent experience with a scripting language similar to PowerShell.




While all of the functions must be presented in the script, only a few require actual logic in order to be implemented.

The input parameters and response format for each function is predefined. All functions receive a set of general parameters whereas those parameters that are *specific* to the function are only passed to it.

## About hash tables

Data is passed to (and returned by) each function using hash table data structures.

A *general* hash table, which includes a common set of data, is passed to all of the functions. A *specific* hashtable, which includes data that is applicable specifically to the function, is passed to functions that require additional data. All functions must return a single hash table that includes a *result* along with any other variables that the function is required to return.

Functions must not return errors; rather, they must throw exceptions in the same way that actual PowerShell errors do. The Adaptable CA driver treats exceptions thrown by a PowerShell function as a fatal error and then halts processing.

### Why Windows PowerShell?

Trust Protection Foundation runs on the Windows Server operating system where PowerShell is already included. PowerShell is also an extremely flexible programming language because it can execute PowerShell *cmdlets* (both native and third party), .NET classes, and any binary that runs on Windows.

By using PowerShell, scripts that are to be used by the Adaptable CA driver can be developed and tested independently of Trust Protection Foundation, making integration simpler.

## Using the sample DigiCert PKI Platform PowerShell script

When you install Trust Protection Foundation, a PowerShell reference sample for use with the DigiCert PKI Platform is included in the \Venafi\Scripts\AdaptableCA\Samples folder. You can use the reference sample to better understand how to configure your own scripts.

The reference sample is a *fully-functional implementation* that lets you use the Adaptable CA driver with the DigiCert PKI Platform. You do not need to know anything about the PowerShell functions to begin using the reference sample. However, you do need to set up your Symantec account correctly and modify a few variables in the script.

**DID YOU KNOW?** CyberArk supports integration with the following Symantec CA offerings:

- Symantec Local Hosting Kit (Symantec LHK): Used for managing legacy, non-publicly trusted certificates.
- Symantec Managed PKI for SSL: Used for managing publicly trusted certificates. For more information, see "Symantec Managed PKI for SSL" on page 195 and visit https://www.symantec.com/ssl-certificates/managed-pki-ssl/.
- DigiCert PKI Platform: Used for managing non-publicly trusted certificates. For more information, visit https://www.symantec.com/products/information-protection/managed-pki-service.

### To use the DigiCert PKI Platform PowerShell script

1. In your DigiCert PKI Platform account, create a certificate profile using Symantec's **Generic Server** template, and then do the following:
   a. Change the enrollment method to **PKI Web Services**.
   b. Add as many instances of the Organization Unit field as you plan to support in your environment.
   c. Add fields for Organization, Locality, State, and Country so that the issued certificates will have a complete Subject DN.
   d. Set the source of all fields to *Webservice Request*.
2. Depending on the certificate type that you want to issue, copy the appropriate script:
   - TLS/SSL certificates – Symantec Managed PKI Service.ps1
   - User (Email Security) certificates – Symantec Managed PKI Service Email.ps1

   from the \Venafi\Scripts\AdaptableCA\Samples directory to the \Venafi\Scripts\AdaptableCA directory.

   **IMPORTANT** You must also copy the PowerShell script to all Trust Protection Foundation servers that are utilized in enrollment, or that are serving up an administration console (e.g. Policy Tree or Aperture).
3. Edit the copied script by doing the following:

a. Verify that the correct DigiCert PKI Platform URL is assigned to the $global:*base_url* variable.

b. Set the value of the $global:*cert_profile_id* variable to the Certificate Profile OID of the certificate profile you created in the first step.

## Configuring the Adaptable CA object

As with other CyberArk CA drivers, you should review the "Getting started: automating certificate enrollment and provisioning" on page 15 section before setting up the Adaptable CA driver.

> **NOTE** Before you attempt to create CA template, device, or application objects, you must enable the *create* permission under the folder where you want to create the new object. For more information, see Permissions Overview in the *Administration Guide*.

### To create and configure a new Adaptable CA template object

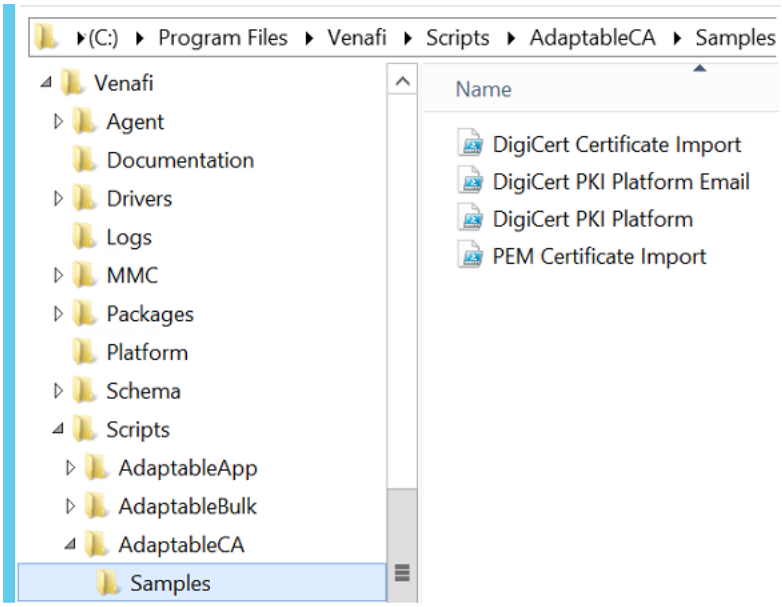
1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. From the **Tree** drop-down menu, click **Policy**.
3. In the **Policy** tree, select the folder where you want to create the CA Template object, and then click **Add**.
4. Click **CA Template**, then select **Adaptable** to create it.
5. In the **CA Name** box, type a name for the new Adaptable object.
6. Under **Connection**, do the following:
   a. Using the **Username Credential** and **Certificate Credential** fields, select one or both credentials.

      If your organization requires two-factor authentication, then specify both credentials.

   b. (Optional) (Conditional) If you need to select another credential, then from the **Secondary Credential** field, select a username, certificate, password, or CyberArk credential object.

      > **TIP** Use this option to avoid having to hard code additional credentials in your script or having to utilize other solutions outside of Trust Protection Foundation.
7. Select your Windows PowerShell script from the **PowerShell Script** list.

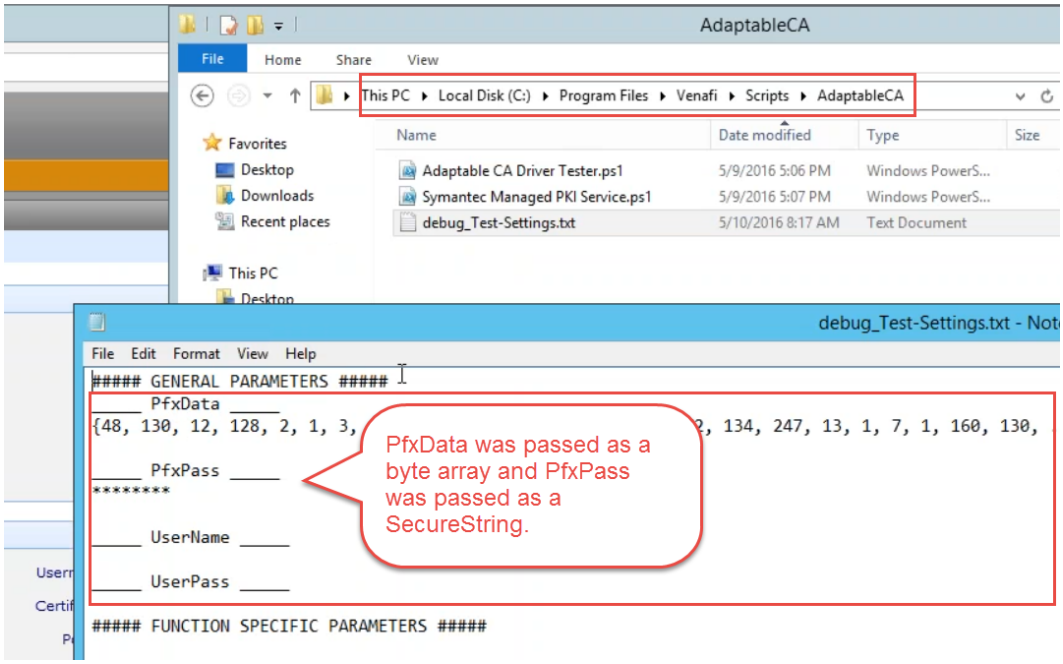
Scripts contained in the Program Files\Venafi\Scripts\AdaptableCA folder appear in this list. For information about creating and modifying Adaptable CA PowerShell scripts, see "About the Adaptable CA PowerShell script" on page 64.




8. To verify that functions are working, click **Validate**.

   **What does Validate do?** When you click *Validate*, Trust Protection Foundation verifies if the required PowerShell functions are present in the selected PowerShell script and then executes the Test-Settings function. In addition, if you had obtained the testing utility from CyberArk Support (see "Adaptable CA prerequisites" on page 63), then the test script would also write a file for every method that gets called to show what data is getting returned. For example, if you had set the certificate credential, Trust Protection Foundation would have generated a text file showing the data that was

sent to the test script:




9. Under **Options**, configure certificate replacement (sometimes called *reissuance*) by doing the following:

   a. In the **Renewal Window (days)** field, specify how many days prior to the expiration of an existing certificate that the certificate should be renewed.

      Any certificates that fall outside of the renewal window are handled through a replacement or reissuance.

   b. To enable renewal *without changing the original expiration date*, check **Allow Reissuance**.

   c. (Optional) Select **When script is updated, fix related certificate errors** if you want Trust Protection Foundation to fix certificates affected by changes to the associated PowerShell script.

      For more information about how changes to scripts can affect certificate enrollment, see "Protecting against unapproved changes to Adaptable CA scripts" on page 73.

10. (Optional) If you want to enhance troubleshooting capabilities of your Adaptable Flow, select the **Enable Debug Logging** check box.

    For information about how enabling this option works with the PowerShell script, see in the Adaptable Flow PowerShell script reference.

11. Click **Add / Remove** to select the custom fields you want to include.

    Data related to the custom fields you select is passed using the functions defined in the PowerShell script.

    For more information about custom fields, see Working with custom fields in the *Trust Protection Foundation Administration Guide*.

12. (Optional) If your application will connect to the CyberArkWeb SDK, then complete the **WebSDK OAuth Token Configuration** settings:

| Setting | Description |
|---|---|
| OAuth Token Application ID | Enter the application ID of the *API application integration* you should have created previously, as described in "Adaptable CA prerequisites" on page 63. |
| OAuth Token Credential | Select the username credential of the service account that has been granted access to the Client ID of the API Application. See "Adaptable CA prerequisites" on page 63.<br><br>In this context, the username credential identifies the user (identity) for whom the token is being requested. It also verifies whether you have the required permissions within your organization to enable the script to authenticate as the selected user. This security measure prevents users from impersonating another user. |
| OAuth Token Scope | (Optional) Enter one or more of the scopes assigned to your API application. For example, Certificates: Manage. Leave this field blank if you want to include all defined scopes. |

| Setting | Description |
| --- | --- |
| | To learn more about scopes and restrictions, see Token Auth scope in the Developer's Guide. |

**NOTE** If your application is not connecting to the Web SDK, leave all of these fields blank.

13. (Conditional) If your script includes customized fields, enter the desired static text or macro commands.

    Refer to the sample in the topic, Example: creating a ServiceNow incident for expiring certificates.

    For more information about CyberArk's macros, see Macro commands in the *Macro Guide*.

14. When you're finished, click **Save**.

15. **IMPORTANT** If you make changes to the PowerShell script used by an Adaptable CA, you must open the corresponding CA object and click **Save** to force the driver to re-read the script. Typically, the updated script becomes active in less than 60 seconds after saving the channel object.

16. Click **Save**.

## What's next?

After you create a CA object, you can select it from the Policy tree, and then view important information and manage various settings.

- Click the **General** tab to view and modify log and permissions settings.
  - Click the **Log** sub-tab to view any logged events that are triggered by the template object.

    **IMPORTANT** You must have the *Read* permission to view the Log tab.

    For more information about options found on the Log tab, see Viewing log events.

  - On the **Permissions** sub-tab, you can configure the users or groups to whom you want to grant permissions to the new template object.

    Consider managing object permissions via parent objects so that you can take advantage of inheritance. For more information, see Permission inheritance and flow down.

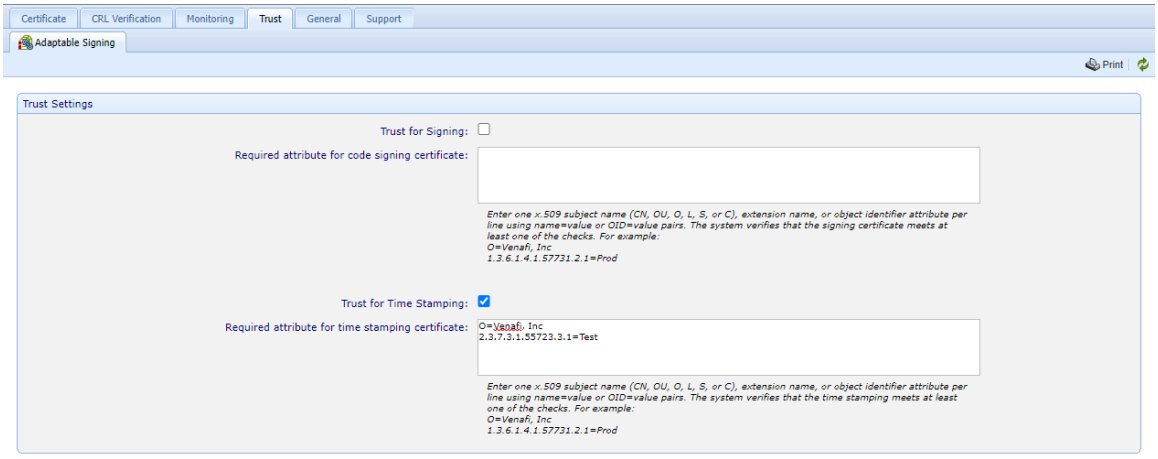
## Configuring certificate trust settings for code signing and time stamping

CyberArk Trust Protection Foundation allows you to configure Adaptable CA using code signed PowerShell scripts. This eliminates the need to approve scripts if they are modified, which removes the operational downtime. The verification process includes verifying script signatures, chaining to a trusted root certificate authority, and optionally verifying the attributes of signing certificates.

Configuring trust for signing or trust for time stamping indicates that you trust all certificates issued by the selected root and its subordinate authorities.

### To configure the root certificate trust settings in the Roots tree

1. From the Certificate Manager - Self-Hosted menu bar, click **Policy Tree**.
2. From the Tree drop-down menu, click **Root**.
3. In the Root tree, select a root certificate.
4. (Optional) Select whether to trust certificates issued by the root and its subordinate certificate authorities for code signing by checking the **Trust for Signing** checkbox.
    a. Enter the required attributes for code signing certificate.
        - Enter only one attribute per line.
        - Attributes include x.509 subject name (CN, OU, L, S, or C), extension name, or object identifier. Enter the attributes using name=value or OID=value pairs.
        - Wildcards are supported.
5. (Optional) Select whether to trust certificates issued by the root and its subordinate certificate authorities for time stamping by checking the **Trust for Time Stamping** checkbox.
    a. Enter the required attributes for time stamping certificate.
        - Enter only one attribute per line.
        - Attributes include x.509 subject name (CN, OU, L, S, or C), extension name, or object identifier. Enter the attributes using name=value or OID=value pairs.
        - Wildcards are supported.




6. Click **Save**.

## Protecting against unapproved changes to Adaptable CA scripts

CyberArk's Adaptable drivers–such as Adaptable CA, Adaptable Application, Adaptable Bulk Provisioning, and Adaptable Log Channel–rely on PowerShell scripts stored in the *\Venafi\Scripts\AdaptableDriverName* directory on Trust Protection Foundation servers. To ensure the integrity of these scripts, Trust Protection Foundation supports signed scripts.

Because other people might have access to the server that is running Trust Protection Foundation, they could modify your PowerShell scripts without your knowledge, either accidentally or intentionally. However, with signed scripts, any modifications made to the script will result in a failed signature validation, and the script will not be executed.

To protect against unapproved changes to your scripts, Trust Protection Foundation monitors PowerShell script files that are being used by existing Adaptable objects. If a new script is used or a PowerShell script is modified on the file system, Trust Protection Foundation displays a warning and you'll need to re-validate the script.

This security feature helps to prevent potentially harmful modifications to your scripts from being run.

> **IMPORTANT** Because of this security feature, following an upgrade to Trust Protection Foundation, you must take specific steps on all existing Adaptable objects in order for them to be re-enabled. Refer to the documentation for each Adaptable driver for details.

When you encounter an error stating that the associated Adaptable CA PowerShell script has been modified, you'll need to open the associated CA template, run validation again (in the **Connection** box), and then re-save the template settings. When you do, the driver is then re-enabled and you can continue using the script.

After you've verified and approved the changes made to the script, Trust Protection Foundation can also be configured to automatically retry all failed issues caused by script changes. This feature is available for the Adaptable CA and Adaptable Application drivers.

### To re-enable an Adaptable CA driver following a change to its associated PowerShell script

1. Open and review the associated scripts to verify that they contain only approved changes.
2. Open Policy Tree and navigate to the associated Adaptable CA template object.
3. In the **Connection** box, click **Validate**.
4. (Optional) Select **When script is updated, fix related certificate errors** if you want Trust Protection Foundation to fix the failed enrollment of affected certificates automatically.

   > **DID YOU KNOW?** Enabling the *When script is updated...* feature instructs Trust Protection Foundation to retry all failed issues automatically once you have verified and approved the changes made to the script. How you approve scripts is done differently for each Adaptable driver.
   >
   > **Example:**
   >
   > Suppose one of your Adaptable CA scripts is modified. You see a warning repeated across ten certificate renewal requests. They are all using the same Adaptable CA template, which depends on the modified script file. The driver blocked processing when it detected script changes, so the renewal process failed. You open the script file to verify that the changes are safe and accurate. Then you open the associated Adaptable CA template, run another validation, set **When script is updated...** to **Yes**, and then save the template again. Because you enabled the option, all ten certificates are retried automatically and processing completes successfully. If you hadn't enabled this setting, you would need to go to each of the ten certificates and rerun certificate renewals manually.
   >
   > The time required for the automated fix to be completed depends on the number of issues. But Trust Protection Foundation fixes the issues as quickly as it can.

5. When you're finished, click **Save**.

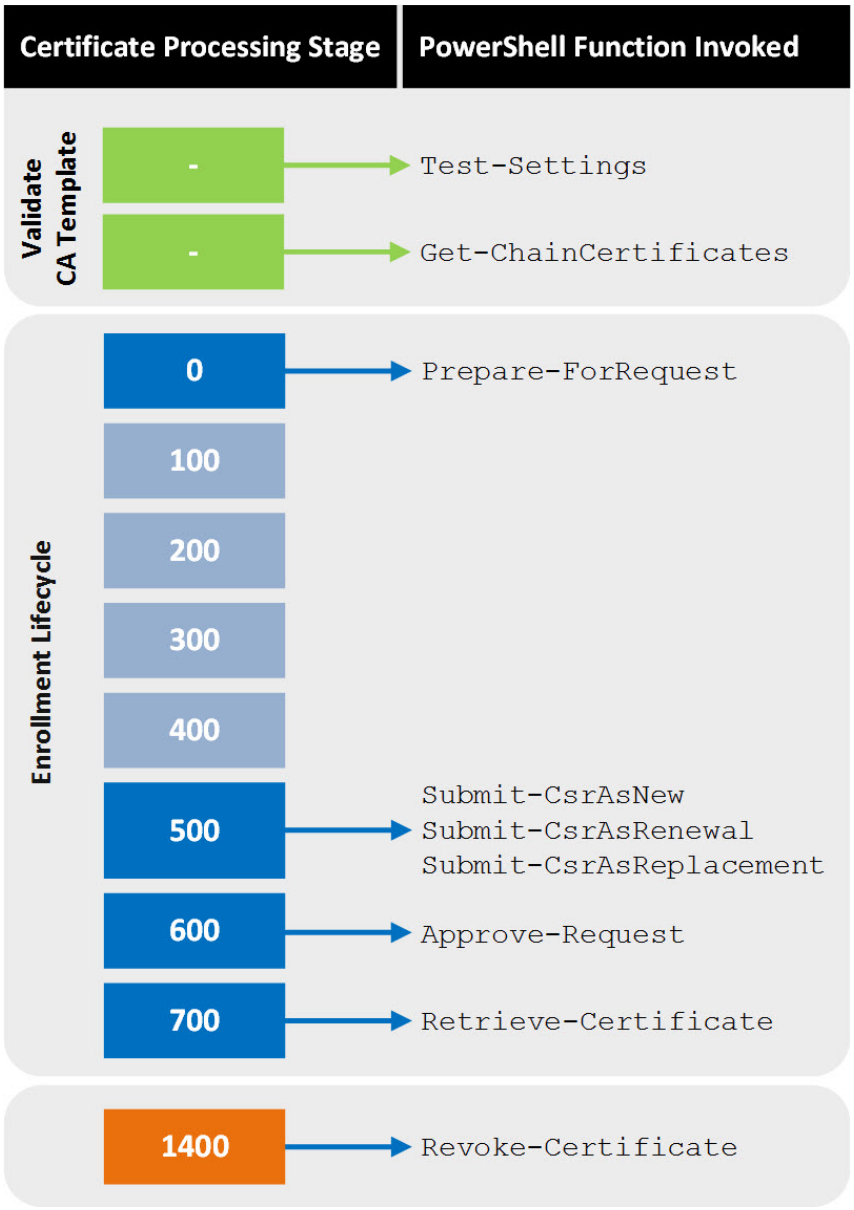
## PowerShell script reference for Adaptable CA

This section contains all available PowerShell functions for use with the Adaptable CA driver. PowerShell scripts are stored in the Program Files\Venafi\Scripts\AdaptableCA folder. However, the sample DigiCert PKI Platform.ps1 is included in the \Venafi\Scripts\AdaptableCA\Samples folder.

For more information about the sample script, see "Using the sample DigiCert PKI Platform PowerShell script" on page 65.

> **NOTE** To work effectively with any CyberArk adaptable solution, you must have some working knowledge of PowerShell scripting, or you must have equivalent experience with a scripting language similar to PowerShell.

The following diagram shows which PowerShell functions are invoked at which stage in the certificate lifecycle:




For information about processing stages, see "About certificate lifecycle management" on page 42.

**BEST PRACTICE** When customizing (or creating a new) PowerShell script, keep the following security best practices in mind:

- Avoid hard-coding credentials into your PowerShell scripts.
- Only include code in functions that relate to the task they are designated to perform.
- Scripts should not do anything that could alter the integrity or availability of the local Windows system (the system that is hosting Trust Protection Foundation).

## About debug logging

When a user has requested debug logging by checking **Enable Debug Logging** for Adaptable FlowAdaptable CAAdaptable SSH Key Discovery, the driver sets a global variable called $DEBUG_FILE whenever it executes a PowerShell function. So your PowerShell script should reference the value of the $DEBUG_FILE variable to decide whether or not to log information for troubleshooting purposes. The value the driver assigns to the $DEBUG_FILE variable is a recommended file path name on the Trust Protection Foundation server for use when logging events to a file. The file name is designed to be unique to the instance of the Adaptable component so as to avoid conflicts when multiple scripts are running at the same time and writing to the log file. If the recommended file name is used, the resulting log file appears in the `<Venafi Home>\Logs` directory by default (e.g. `C:\Program Files\Venafi\Logs`).

For information about where Enable Debug Logging is configured for Adaptable FlowAdaptable CA, see "Configuring the Adaptable CA object" on page 67.

## Variable reference

| Variable Name | Data Type | Description |
|---|---|---|
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |

| Variable Name | Data Type | Description |
|---|---|---|
| CertObjDN | String | CyberArk distinguished name (DN) of the certificate object in the policy tree |
| CustomFields | Hashtable | Hash table keys are Custom Field labels and the values are strings (single-valued) or string arrays (multi-valued). |
| OAuthAccessToken | String | OAuth token passed to the PowerShell script. Tokens are passed only after you've provided the required information in the *WebSDK OAuth Token Configuration* settings of the Adaptable object. See "Configuring the Adaptable CA object" on page 67.<br><br>How it works<br><br>When used, each time your script is called, Trust Protection Foundation requests a new token automatically. Each new token is associated with both the specified user and with the referenced application ID. After the script finishes running, Trust Protection Foundation revokes the token automatically. To get started, see "Adaptable CA prerequisites" on page 63. |
| PfxData | Byte Array | PKCS#12 keystore containing client certificate and private key for authenticating with the CA; this and the PfxPass are used together to instantiate an X509Certificate2 object for client certificate authentication. |
| PfxPass | SecureString | Password for access to the private key of the PfxData PKCS#12 |
| UserName | String | User name for authenticating with the CA |
| UserPass | String | Password for authenticating with the CA |
| WebSdkUrl | String | String representing the fully-qualified domain name to the WebSDK of your Trust Protection Foundation server. For information on where this variable data is set, see Trust Protection Foundation server configuration. |

## Approve-Request function

The Approve-Request function is called at processing stage 600 and is used to automate the approval of a certificate request that was made by Submit-CsrAsNew, Submit-CsrAsRenewal, or Submit-CsrAsReplacement.

You do not need to implement logic for this function for those CAs that can be configured to automatically issue certificates that are submitted by specific credentials or by specific methods (API).

| Variable Name | Data Type | Description |
|---|---|---|
| SubjAltNames | Hashtable | Hash table keyed by SAN type; values are string arrays. |
| SubjectDN | Hashtable | The requested subject distinguished name (DN) as a hash table; OU is a string array; all others are strings. |
| TransId | String | Text used by the CA to uniquely identify the certificate request submitted previously.<br><br>**NOTE** When configuring the Adaptable CA, do not hard-code the Transaction ID (TransId) field.<br>■ If a transaction Id is required, generate a unique value for each request.<br>■ Alternately, leave the field empty to allow the system to manage it.<br><br>Hard-coding even a single, fixed transaction ID can cause performance degradation and unintended behavior. |

| Variable Name | Data Type | Description |
|---|---|---|
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| CertObjDN | String | CyberArk distinguished name (DN) of the certificate object in the policy tree |

| Variable Name | Data Type | Description |
|---|---|---|
| CustomFields | Hashtable | Hash table keys are Custom Field labels and the values are strings (single-valued) or string arrays (multi-valued). |
| OAuthAccessToken | String | OAuth token passed to the PowerShell script. Tokens are passed only after you've provided the required information in the *WebSDK OAuth Token Configuration* settings of the Adaptable object. See "Configuring the Adaptable CA object" on page 67.<br><br>How it works<br><br>When used, each time your script is called, Trust Protection Foundation requests a new token automatically. Each new token is associated with both the specified user and with the referenced application ID. After the script finishes running, Trust Protection Foundation revokes the token automatically. To get started, see "Adaptable CA prerequisites" on page 63. |
| PfxData | Byte Array | PKCS#12 keystore containing client certificate and private key for authenticating with the CA; this and the PfxPass are used together to instantiate an X509Certificate2 object for client certificate authentication. |
| PfxPass | SecureString | Password for access to the private key of the PfxData PKCS#12 |
| UserName | String | User name for authenticating with the CA |
| UserPass | String | Password for authenticating with the CA |
| WebSdkUrl | String | String representing the fully-qualified domain name to the WebSDK of your Trust Protection Foundation server. For information on where this variable data is set, see Trust Protection Foundation server configuration. |

| Return | Data Type | Description |
|---|---|---|
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state. |
| TransId | String | Text used by the CA to uniquely identify the certificate request submitted previously. |

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Get-ChainCertificates function

The Get-ChainCertificates function returns the root and intermediate certificate chain for the end-entity certificates to be issued by the certificate authority (CA).

The Get-ChainCertificates function is called when the CA template is validated and any certificates that are returned are added to the Adaptable CA template automatically. This helps to ensure that the proper chain is built for the certificates that are enrolled.

| Variable Name | Data Type | Description |
|---|---|---|
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| CertObjDN | String | CyberArk distinguished name (DN) of the certificate object in the policy tree |
| CustomFields | Hashtable | Hash table keys are Custom Field labels and the values are strings (single-valued) or string arrays (multi-valued). |
| OAuthAccessToken | String | OAuth token passed to the PowerShell script. Null is always passed as a value, because this function is only used for CA settings validation. |
| PfxData | Byte Array | PKCS#12 keystore containing client certificate and private key for authenticating with the CA; this and the PfxPass are used together to instantiate an X509Certificate2 object for client certificate authentication. |
| PfxPass | SecureString | Password for access to the private key of the PfxData PKCS#12 |

| Variable Name | Data Type | Description |
|---|---|---|
| UserName | String | User name for authenticating with the CA |
| UserPass | String | Password for authenticating with the CA |
| WebSdkUrl | String | String representing the fully-qualified domain name to the WebSDK of your Trust Protection Foundation server. Null is always passed as a value, because this function is only used for CA settings validation. For information on where this variable data is set, see Trust Protection Foundation server configuration. |

| Return | Data Type | Description |
|---|---|---|
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state. |
| PKCS7 | String | A collection that includes all of the CA certificates in the issuing chain. |

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Prepare-ForRequest function

This function is called at the very beginning of the certificate lifecycle (at stage 0). It can be used to complete any prerequisite tasks that are needed before an actual request is submitted to the CA.

| Variable Name | Data Type | Description |
|---|---|---|
| CustomExtensions | Hashtable | Custom certificate extensions being requested; Hashtable keys are 1.3.6.1.4.1.311.25.2 (SID extension). Values are strings. |
| KeySize | String | The integer key size to be used if the CA creates the key pair. |
| SubjectDN | Hashtable | Subject Distinguished Name being requested; Hashtable keys are CN (Common Name), OU (Organizational Units), O (Organization), L (City/Locality), ST (State/Province), and C (Country). Values are strings or string arrays (only when there are multiple OUs). |
| SubjAltNames | Hashtable | Subject Alternative Names being requested; Hashtable keys are DNS, Email, IP, URI, and UPN. Values are strings (if a single value) or string arrays (if multi-valued). |

| Variable Name | Data Type | Description |
|---|---|---|
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| CertObjDN | String | CyberArk distinguished name (DN) of the certificate object in the policy tree |
| CustomFields | Hashtable | Hash table keys are Custom Field labels and the values are strings (single-valued) or string arrays (multi-valued). |
| OAuthAccessToken | String | OAuth token passed to the PowerShell script. Tokens are passed only after you've provided the required information in the *WebSDK OAuth Token Configuration* settings of the Adaptable object. See "Configuring the Adaptable CA object" on page 67.<br><br>How it works<br><br>When used, each time your script is called, Trust Protection Foundation requests a new token automatically. Each new token is associated with both the specified user and with the referenced application ID. After the script finishes running, Trust Protection Foundation revokes the token automatically. To get started, see "Adaptable CA prerequisites" on page 63. |
| PfxData | Byte Array | PKCS#12 keystore containing client certificate and private key for authenticating with the CA; this and the PfxPass are used together to instantiate an X509Certificate2 object for client certificate authentication. |
| PfxPass | SecureString | Password for access to the private key of the PfxData PKCS#12 |

| Variable Name | Data Type | Description |
|---|---|---|
| UserName | String | User name for authenticating with the CA |
| UserPass | String | Password for authenticating with the CA |
| WebSdkUrl | String | String representing the fully-qualified domain name to the WebSDK of your Trust Protection Foundation server. For information on where this variable data is set, see Trust Protection Foundation server configuration. |

| Return | Data Type | Description |
|---|---|---|
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state. |

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Retrieve-Certificate function

Implementing logic for the Retrieve-Certificate function is required unless the Submit-CsrAsNew, Submit-CsrAsRenewal, and Submit-CsrAsReplacement functions are all written to return the certificate issued by the CA.

When implemented, this function must attempt to retrieve a certificate from the CA. It is called at Processing Stage 700 and could return a *Pending* status to indicate that the driver should retry the operation again later. When the certificate issued by the CA is available, the function must return a status of *Issued* and include the certificate in the same response.

> **NOTE** The Retrieve-Certificate function is not called when the Submit-CsrAsNew, Submit-CsrAsRenewal, or Submit-CsrAsReplacement functions return a certificate in the Pkcs7 or Pkcs12 parameters. Also, this function must return Pkcs7 or Pkcs12 unless they were returned by the Submit function.
>
> Refer to the "Retrieve-Certificate function" above table below.

| Variable Name | Data Type | Description |
|---|---|---|
| CertId | String | Text that uniquely identifies the certificate issued by the CA. |

| Variable Name | Data Type | Description |
|---|---|---|
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| CertObjDN | String | CyberArk distinguished name (DN) of the certificate object in the policy tree |
| CustomFields | Hashtable | Hash table keys are Custom Field labels and the values are strings (single-valued) or string arrays (multi-valued). |
| OAuthAccessToken | String | OAuth token passed to the PowerShell script. Tokens are passed only after you've provided the required information in the *WebSDK OAuth Token Configuration* settings of the Adaptable object. See "Configuring the Adaptable CA object" on page 67.<br><br>How it works<br><br>When used, each time your script is called, Trust Protection Foundation requests a new token automatically. Each new token is associated with both the specified user and with the referenced application ID. After the script finishes running, Trust Protection Foundation revokes the token automatically. To get started, see "Adaptable CA prerequisites" on page 63. |
| PfxData | Byte Array | PKCS#12 keystore containing client certificate and private key for authenticating with the CA; this and the PfxPass are used together to instantiate an X509Certificate2 object for client certificate authentication. |
| PfxPass | SecureString | Password for access to the private key of the PfxData PKCS#12 |

| Variable Name | Data Type | Description |
|---|---|---|
| UserName | String | User name for authenticating with the CA |
| UserPass | String | Password for authenticating with the CA |
| WebSdkUrl | String | String representing the fully-qualified domain name to the WebSDK of your Trust Protection Foundation server. For information on where this variable data is set, see Trust Protection Foundation server configuration. |

| Return | Data Type | Description |
|---|---|---|
| DiscardPrivateKeyAndCsr | Bool | A boolean that when $true tells the driver to discard stored assets that will not match the certificate being returned |
| EncryptPass | String | The password used to encrypt the private key or PKCS#12. |
| Pkcs7 | Array | A collection that includes all of the CA certificates in the issuing chain. |
| Pkcs12 | Array | A collection that includes the issued certificate, its private key, and (optionally) all of the CA certificates in the chain.<br><br>When PKCS#12 is returned, EncryptPass must also be returned and Pkcs7 and PrivKeyPem are ignored.<br><br>Also, if the PrivKeyPem data is a password-encrypted private key, EncryptPass must be returned. |
| PrivKeyPem | String | An OpenSSL or PKCS#8 private key in PEM format. |

| Return | Data Type | Description |
|---|---|---|
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state. |
| Status | String | Shows "Pending" or "Issued" to indicate whether or not the certificate has been issued by the CA. |
| TransId | String | Text used by the CA to uniquely identify the certificate request submitted previously.<br><br>**NOTE** When configuring the Adaptable CA, do not hard-code the Transaction ID (TransId) field.<br>■ If a transaction Id is required, generate a unique value for each request.<br>■ Alternately, leave the field empty to allow the system to manage it.<br><br>Hard-coding even a single, fixed transaction ID can cause performance degradation and unintended behavior. |

**IMPORTANT** If Pkcs12 or PrivKeyPem are returned, DiscardPrivateKeyAndCsr is automatically set to $true and cannot be overridden.

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Import-Certificates function

Use this function to import certificates and keys into Trust Protection Foundation.

| Variable Name | Data Type | Description |
|---|---|---|
| IncludeRevoked | Boolean | Indicates whether the job says to include revoked certificates in the response |
| IncludeExpired | Boolean | Indicates whether the job says to include expired certificates in the response |
| LastJobStarted | DateTime | Indicates when the previous import job started; can be used to select only certificates issues after |
| LastProcessedId | String | Transaction ID of the last certificate that was previously imported by the job; can be used to select only certificates issued after |
| CustomFields | Array | Names of all custom fields that are currently defined for certificates |

| Variable Name | Data Type | Description |
|---|---|---|
| AuxPass | String | The password for the AuxUser (username credential), AuxPfxData (certificate credential), or independent (password credential) |
| AuxPfxData | Byte Array | PKCS#12 keystore containing a client certificate and private key for secondary authentication purposes |
| AuxUser | String | Username paired with AuxPass for secondary authentication purposes |
| PfxData | Byte Array | PKCS#12 keystore containing a client certificate and private key for authentication |
| PfxPass | SecureString | Password required to access the private key in the PfxData |
| UserName | String | Username to use when authenticating with the CA or other certificate repository |
| UserPass | String | Password to use when authenticating with the CA or other certificate repository |

| Return | Data Type | Description |
|---|---|---|
| **Certificates** | Array | An array of hashtables containing certificates, keys, and corresponding metadata |
| Attributes | Hashtable | Additional Config attributes name/value pairs to assign to the certificate object |
| ContactEmail | String | Email address of the certificate contact; will be used to assign a contact identity to the certificate or assigned to the Internet Email Address attribute if no identity is found |
| CustomFields | Hashtable | Custom Field name/value pairs to assign to the certificate object |
| CustomName | String | Name to use for the certificate object in the policy tree; if not specified certificate object will use CN, or first DNS SAN, or first OU, or serial number (in that order) |
| Name | String | (Required) Identifier that will appear in the log event if the certificate fails to import |
| Password | String | Password needed to decrypt password encrypted private keys and PKCS#12 keystores |
| PEM | String | (required) The raw certificate expressed in PEM format or as a base64 encoded PKCS#12 keystore |
| RevocationDate | String | The date on which the certificate was revoked expressed as YYYY-MM-DD |
| TransactionId | String | The unique ID assigned to the certificate by the CA or other certificate repository |
| **LastProcessedId** | String | Transaction ID of the last certificate in the Certificates hash table; can be used to tell the job where to start the next time it runs |
| **Result** | String | Either "Success" or "NotUsed" to indicate a non-error completion state, or "Incomplete" to have the job automatically re-run later to complete the import |
| **RunAfter** | Integer | Number of hours after which script will be executed again when Result="Incomplete" |

## Revoke-Certificate function

While not required, implementing logic for this function is strongly recommended as revocation is functionality that every CA is expected to support, and a standard feature of Trust Protection Foundation. This function is called whenever a user requests that a certificate be revoked and corresponds to Processing Stage 1400.

| Variable Name | Data Type | Description |
|---|---|---|
| CertId | String | Text used by the CA to uniquely identify the certificate being revoked. |
| CertPem | String | The raw X.509 certificate being revoked is in base64-encoded PEM format. |
| ReasonCode | Integer | One of the following RFC 5280 Revocation Reason Codes:<br>0=Unspecified<br>1=Key Compromised<br>2=Affiliation Changed<br>3=Superseded<br>4=Cessation of Operation |
| Comment | String | An optional string that explains why the certificate is being revoked. |

| Variable Name | Data Type | Description |
|---|---|---|
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |

| Variable Name | Data Type | Description |
|---|---|---|
| CertObjDN | String | CyberArk distinguished name (DN) of the certificate object in the policy tree |
| CustomFields | Hashtable | Hash table keys are Custom Field labels and the values are strings (single-valued) or string arrays (multi-valued). |
| OAuthAccessToken | String | OAuth token passed to the PowerShell script. Tokens are passed only after you've provided the required information in the *WebSDK OAuth Token Configuration* settings of the Adaptable object. See "Configuring the Adaptable CA object" on page 67.<br><br>How it works<br><br>When used, each time your script is called, Trust Protection Foundation requests a new token automatically. Each new token is associated with both the specified user and with the referenced application ID. After the script finishes running, Trust Protection Foundation revokes the token automatically. To get started, see "Adaptable CA prerequisites" on page 63. |
| PfxData | Byte Array | PKCS#12 keystore containing client certificate and private key for authenticating with the CA; this and the PfxPass are used together to instantiate an X509Certificate2 object for client certificate authentication. |
| PfxPass | SecureString | Password for access to the private key of the PfxData PKCS#12 |
| UserName | String | User name for authenticating with the CA |
| UserPass | String | Password for authenticating with the CA |
| WebSdkUrl | String | String representing the fully-qualified domain name to the WebSDK of your Trust Protection Foundation server. For information on where this variable data is set, see Trust Protection Foundation server configuration. |

| Return | Data Type | Description |
|---|---|---|
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state. |

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Submit-CsrAsNew function

Implementing logic for this function is mandatory. It must submit a certificate request to a certificate authority (CA) and return an identifier by which the CA uniquely identifies the request and/or the certificate that was issued from it.

The Submit-CsrAsNew function is called at Processing Stage 500 (when a new certificate object is enrolled for the first time) and, with rare exceptions, should submit the PKCS#10 passed to it by the driver to the CA including additional (optional) request options and/or metadata that are needed by the CA.

In cases where a CA returns the certificate immediately in response to a request, this function may return the certificate instead of having to do so using Retrieve-Certificate.

| Variable Name | Data Type | Description |
|---|---|---|
| CertId | String | Text used by the CA to uniquely identify the certificate being renewed. |
| CertPem | String | Raw X.509 Certificate being renewed in base64-encoded PEM forma |
| CustomExtensions | Hashtable | Custom certificate extensions being requested; Hashtable keys are 1.3.6.1.4.1.311.25.2 (SID extension). Values are strings. |
| KeySize | String | The integer key size to be used if the CA creates the key pair. |
| Pkcs10 | String | Certificate Signing Request to be enrolled in base64-encoded PKCS#10 format. |
| SubjectDN | Hashtable | Subject Distinguished Name being requested; Hashtable keys are CN (Common Name), OU (Organizational Units), O (Organization), L (City/Locality), ST (State/Province), and C (Country). Values are strings or string arrays (only when there are multiple OUs). |
| SubjAltNames | Hashtable | Subject Alternative Names being requested; Hashtable keys are DNS, Email, IP, URI, and UPN. Values are strings (if a single value) or string arrays (if multi-valued). |

| Variable Name | Data Type | Description |
|---|---|---|
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| CertObjDN | String | CyberArk distinguished name (DN) of the certificate object in the policy tree |
| CustomFields | Hashtable | Hash table keys are Custom Field labels and the values are strings (single-valued) or string arrays (multi-valued). |
| OAuthAccessToken | String | OAuth token passed to the PowerShell script. Tokens are passed only after you've provided the required information in the *WebSDK OAuth Token Configuration* settings of the Adaptable object. See "Configuring the Adaptable CA object" on page 67.<br><br>How it works<br><br>When used, each time your script is called, Trust Protection Foundation requests a new token automatically. Each new token is associated with both the specified user and with the referenced application ID. After the script finishes running, Trust Protection Foundation revokes the token automatically. To get started, see "Adaptable CA prerequisites" on page 63. |
| PfxData | Byte Array | PKCS#12 keystore containing client certificate and private key for authenticating with the CA; this and the PfxPass are used together to instantiate an X509Certificate2 object for client certificate authentication. |
| PfxPass | SecureString | Password for access to the private key of the PfxData PKCS#12 |

| Variable Name | Data Type | Description |
|---|---|---|
| UserName | String | User name for authenticating with the CA |
| UserPass | String | Password for authenticating with the CA |
| WebSdkUrl | String | String representing the fully-qualified domain name to the WebSDK of your Trust Protection Foundation server. For information on where this variable data is set, see Trust Protection Foundation server configuration. |

| Return | Data Type | Description |
|---|---|---|
| EncryptPass | String | The password used to encrypt the private key or PKCS#12. |
| Pkcs7 | String | A collection that includes the issued certificate and (optionally) includes all of the CA certificates in the chain. |
| Pkcs12 | String | A collection that includes the issued certificate, its private key, and (optionally) all of the CA certificates in the chain.<br><br>When PKCS#12 is returned, EncryptPass must also be returned and Pkcs7 and PrivKeyPem are ignored. |

| Return | Data Type | Description |
|---|---|---|
| PrivKeyPem | String | An OpenSSL or PKCS#8 private key in PEM format.<br><br>When PrivKeyPem is returned, PKCS7 must also be returned.<br><br>Also, if the PrivKeyPem data is a password-encrypted private key, EncryptPass must be returned. |
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state. |
| TransID | String | The identifier used to subsequently approve and/or retrieve the certificate from the CA.<br><br>**NOTE** When configuring the Adaptable CA, do not hard-code the Transaction ID (TransId) field.<br><br>■ If a transaction Id is required, generate a unique value for each request.<br><br>■ Alternately, leave the field empty to allow the system to manage it.<br><br>Hard-coding even a single, fixed transaction ID can cause performance degradation and unintended behavior. |

**IMPORTANT** If Pkcs12 or PrivKeyPem are returned, the private key and CSR currently stored for the certificate object are discarded. This is because they won't match the certificate issued by the CA.

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Submit-CsrAsRenewal function

The Submit-CsrAsRenewal function is basically the same as Submit-CsrAsNew with one important difference: it is called when a certificate is requested to renew an existing certificate that is nearing expiration.

Certificate renewal occurs when the expiring certificate is inside the Renewal Window configured on the CA template. That is, the expiring certificate has fewer days remaining in its Validity Period than the number of days in the Renewal Window.

If the CA does not differentiate between first time certificate requests and renewal of existing certificates, this function should be written to call Submit-CsrAsNew and pass its response back to the driver.

| Variable Name | Data Type | Description |
|---|---|---|
| CertId | String | Text used by the CA to uniquely identify the certificate being renewed. |
| CertPem | String | Raw X.509 Certificate being renewed in base64-encoded PEM forma |
| CustomExtensions | Hashtable | Custom certificate extensions being requested; Hashtable keys are 1.3.6.1.4.1.311.25.2 (SID extension). Values are strings. |
| KeySize | String | The integer key size to be used if the CA creates the key pair. |
| Pkcs10 | String | Certificate Signing Request to be enrolled in base64-encoded PKCS#10 format. |
| SubjectDN | Hashtable | Subject Distinguished Name being requested; Hashtable keys are CN (Common Name), OU (Organizational Units), O (Organization), L (City/Locality), ST (State/Province), and C (Country). Values are strings or string arrays (only when there are multiple OUs). |
| SubjAltNames | Hashtable | Subject Alternative Names being requested; Hashtable keys are DNS, Email, IP, URI, and UPN. Values are strings (if a single value) or string arrays (if multi-valued). |

| Variable Name | Data Type | Description |
|---|---|---|
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| CertObjDN | String | CyberArk distinguished name (DN) of the certificate object in the policy tree |

| Variable Name | Data Type | Description |
|---|---|---|
| CustomFields | Hashtable | Hash table keys are Custom Field labels and the values are strings (single-valued) or string arrays (multi-valued). |
| OAuthAccessToken | String | OAuth token passed to the PowerShell script. Tokens are passed only after you've provided the required information in the *WebSDK OAuth Token Configuration* settings of the Adaptable object. See "Configuring the Adaptable CA object" on page 67.<br><br>How it works<br><br>When used, each time your script is called, Trust Protection Foundation requests a new token automatically. Each new token is associated with both the specified user and with the referenced application ID. After the script finishes running, Trust Protection Foundation revokes the token automatically. To get started, see "Adaptable CA prerequisites" on page 63. |
| PfxData | Byte Array | PKCS#12 keystore containing client certificate and private key for authenticating with the CA; this and the PfxPass are used together to instantiate an X509Certificate2 object for client certificate authentication. |
| PfxPass | SecureString | Password for access to the private key of the PfxData PKCS#12 |
| UserName | String | User name for authenticating with the CA |
| UserPass | String | Password for authenticating with the CA |
| WebSdkUrl | String | String representing the fully-qualified domain name to the WebSDK of your Trust Protection Foundation server. For information on where this variable data is set, see Trust Protection Foundation server configuration. |

| Return | Data Type | Description |
|---|---|---|
| EncryptPass | String | The password used to encrypt the private key or PKCS#12. |
| Pkcs7 | String | A collection that includes the issued certificate and (optionally) includes all of the CA certificates in the chain. |
| Pkcs12 | String | A collection that includes the issued certificate, its private key, and (optionally) all of the CA certificates in the chain.<br><br>When PKCS#12 is returned, EncryptPass must also be returned and Pkcs7 and PrivKeyPem are ignored. |
| PrivKeyPem | String | An OpenSSL or PKCS#8 private key in PEM format.<br><br>When PrivKeyPem is returned, PKCS7 must also be returned.<br><br>Also, if the PrivKeyPem data is a password-encrypted private key, EncryptPass must be returned. |
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state. |
| TransID | String | The identifier used to subsequently approve and/or retrieve the certificate from the CA.<br><br>**NOTE** When configuring the Adaptable CA, do not hard-code the Transaction ID (TransId) field.<br>■ If a transaction Id is required, generate a unique value for each request.<br>■ Alternately, leave the field empty to allow the system to manage it.<br><br>Hard-coding even a single, fixed transaction ID can cause performance degradation and unintended behavior. |

**IMPORTANT** If Pkcs12 or PrivKeyPem are returned, the private key and CSR currently stored for the certificate object are discarded. This is because they won't match the certificate issued by the CA.

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Submit-CsrAsReplacement function

The Submit-CsrAsReplacement function is basically the same as the Submit-CsrAsNew function but with one important difference: it is called when a certificate is requested to replace an existing certificate that is *not* nearing expiration.

Certificate replacement–or reissuance as it is sometimes called–occurs when the expiring certificate is outside of the Renewal Window that has been configured on the CA template. That is, the expiring certificate has more days remaining in its Validity Period than the number of days in the Renewal Window. If the CA does not differentiate between first-time certificate requests and replacement of existing certificates, this function should be written to call Submit-CsrAsNew and pass its response back to the driver.

| Variable Name | Data Type | Description |
|---|---|---|
| CertId | String | Text used by the CA to uniquely identify the certificate being renewed. |
| CertPem | String | Raw X.509 Certificate being renewed in base64-encoded PEM forma |
| CustomExtensions | Hashtable | Custom certificate extensions being requested; Hashtable keys are 1.3.6.1.4.1.311.25.2 (SID extension). Values are strings. |
| KeySize | String | The integer key size to be used if the CA creates the key pair. |
| Pkcs10 | String | Certificate Signing Request to be enrolled in base64-encoded PKCS#10 format. |
| SubjectDN | Hashtable | Subject Distinguished Name being requested; Hashtable keys are CN (Common Name), OU (Organizational Units), O (Organization), L (City/Locality), ST (State/Province), and C (Country). Values are strings or string arrays (only when there are multiple OUs). |
| SubjAltNames | Hashtable | Subject Alternative Names being requested; Hashtable keys are DNS, Email, IP, URI, and UPN. Values are strings (if a single value) or string arrays (if multi-valued). |

| Variable Name | Data Type | Description |
|---|---|---|
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| CertObjDN | String | CyberArk distinguished name (DN) of the certificate object in the policy tree |
| CustomFields | Hashtable | Hash table keys are Custom Field labels and the values are strings (single-valued) or string arrays (multi-valued). |
| OAuthAccessToken | String | OAuth token passed to the PowerShell script. Tokens are passed only after you've provided the required information in the *WebSDK OAuth Token Configuration* settings of the Adaptable object. See "Configuring the Adaptable CA object" on page 67.<br><br>How it works<br><br>When used, each time your script is called, Trust Protection Foundation requests a new token automatically. Each new token is associated with both the specified user and with the referenced application ID. After the script finishes running, Trust Protection Foundation revokes the token automatically. To get started, see "Adaptable CA prerequisites" on page 63. |
| PfxData | Byte Array | PKCS#12 keystore containing client certificate and private key for authenticating with the CA; this and the PfxPass are used together to instantiate an X509Certificate2 object for client certificate authentication. |
| PfxPass | SecureString | Password for access to the private key of the PfxData PKCS#12 |

| Variable Name | Data Type | Description |
|---|---|---|
| UserName | String | User name for authenticating with the CA |
| UserPass | String | Password for authenticating with the CA |
| WebSdkUrl | String | String representing the fully-qualified domain name to the WebSDK of your Trust Protection Foundation server. For information on where this variable data is set, see Trust Protection Foundation server configuration. |

| Return | Data Type | Description |
|---|---|---|
| EncryptPass | String | The password used to encrypt the private key or PKCS#12. |
| Pkcs7 | String | A collection that includes the issued certificate and (optionally) includes all of the CA certificates in the chain. |
| Pkcs12 | String | A collection that includes the issued certificate, its private key, and (optionally) all of the CA certificates in the chain.<br><br>When PKCS#12 is returned, EncryptPass must also be returned and Pkcs7 and PrivKeyPem are ignored. |

| Return | Data Type | Description |
|---|---|---|
| PrivKeyPem | String | An OpenSSL or PKCS#8 private key in PEM format.<br><br>When PrivKeyPem is returned, PKCS7 must also be returned.<br><br>Also, if the PrivKeyPem data is a password-encrypted private key, EncryptPass must be returned. |
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state. |
| TransID | String | The identifier used to subsequently approve and/or retrieve the certificate from the CA.<br><br>**NOTE** When configuring the Adaptable CA, do not hard-code the Transaction ID (TransId) field.<br><br>■ If a transaction Id is required, generate a unique value for each request.<br><br>■ Alternately, leave the field empty to allow the system to manage it.<br><br>Hard-coding even a single, fixed transaction ID can cause performance degradation and unintended behavior. |

**IMPORTANT** If Pkcs12 or PrivKeyPem are returned, the private key and CSR currently stored for the certificate object are discarded. This is because they won't match the certificate issued by the CA.

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Test-Settings function

You can use the Test-Settings function to verify that Trust Protection Foundation can connect to (and authenticate with) the CA using the supplied credentials.

Use this function to do the following:

- Verify connectivity with the remote management interface of the CA.
- Exercise the credentials assigned to the CA template in Trust Protection Foundation to verify

successful authentication.

- (Recommended) Execute a benign API method (query) to test access to the API itself.

| Variable Name | Data Type | Description |
|---|---|---|
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| CertObjDN | String | CyberArk distinguished name (DN) of the certificate object in the policy tree |
| CustomFields | Hashtable | Hash table keys are Custom Field labels and the values are strings (single-valued) or string arrays (multi-valued). |
| OAuthAccessToken | String | OAuth token passed to the PowerShell script. Null is always passed as a value, because this function is only used for CA settings validation. |
| PfxData | Byte Array | PKCS#12 keystore containing client certificate and private key for authenticating with the CA; this and the PfxPass are used together to instantiate an X509Certificate2 object for client certificate authentication. |
| PfxPass | SecureString | Password for access to the private key of the PfxData PKCS#12 |
| UserName | String | User name for authenticating with the CA |
| UserPass | String | Password for authenticating with the CA |
| WebSdkUrl | String | String representing the fully-qualified domain name to the WebSDK of your Trust Protection Foundation server. Null is always passed as a value, because this function is only used for CA settings validation. For information on where this variable data is set, see Trust Protection Foundation server configuration. |

| Return | Data Type | Description |
|---|---|---|
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state. |

For information about processing stages, see "About certificate lifecycle management" on page 42.

# Integrating Amazon Certificate Manager with Trust Protection Foundation

Use Trust Protection Foundation's AWS Certificate Manager (ACM) driver to integrate with the Amazon Web Services (AWS) ecosystem to better secure your machine identities.

ACM is designed to simplify certificate acquisition and administrative processes by generating and storing keys, CSRs and certificates internally.

You can use the Trust Protection Foundation ACM driver to manage both public and private (external and internal) certificates across multiple regions.

## Before you start

Before you can issue *private* certificates, you need to create a private CA within each region where you want to issue and revoke private digital certificates.

When you request a *public* ACM certificate, you only need to specify valid names: a Common Name, and any number of DNS Subject Alternative Names. Other settings that are generally relevant to key and CSR generation are not applicable when using ACM. To learn more about ACM, see https://aws.amazon.com/certificate-manager/.

### Did you know?

> **DID YOU KNOW?** CyberArk Trust Protection Foundation™ can request and retrieve certificates from ACM, and also from Amazon Private CAs. This enables your organization to support those certificates using essentially the same processes you have used for other certificate authorities.

**DID YOU KNOW?** Certificates issued by the Amazon CA are also renewed by the Amazon CA automatically *if they are in use by an AWS application*. Therefore, it isn't necessary for Trust Protection Foundation to do any further lifecycle management for those certificates (SSL/TLS Validation should be used to ensure that Amazon renewed the certificate automatically, as expected).

However, if the Amazon CA renews the certificate automatically, the renewed certificate will then have to be imported into Trust Protection Foundation. Amazon starts the renewal process 60 days before the certificate expires; then, in some cases, domain control validation (DCV) processes can delay its completion by a few days. When Trust Protection Foundation attempts to renew the certificate, Amazon returns the *current version* of the certificate–the one with exactly the same CN and DNS SANs–if it exists.

If this chain of events occurs *before* 60 days from expiration, there will be no change in the certificate despite having completed the enrollment lifecycle; but if it happens *after* Amazon has completed the renewal of the certificate, then Trust Protection Foundation retrieves the *new certificate*.

To avoid this issue, you should adjust your ACM-issued certificate settings this way:

- Set the **Disable Automatic Renewal** setting to **No** (disabled).
- Add an additional week to the **Renewal Window** (about 7 days, or approximately 53 days total) to give the DCV process time to complete before the renewal process starts.

For more information, refer the following Amazon FAQ:

https://aws.amazon.com/certificate-manager/faqs

## Ready to go?

From a high level, here are the key steps we'll take to integrate AWS with CyberArk Trust Protection Foundation:

- "Step 1: Verify that your Amazon account is ready to go" on the next page
- "Step 2: Set up an Amazon Private CA (Optional)" on page 107 (Optional)
- "Step 3: Create an Amazon Credential" on page 107 in CyberArk Trust Protection Foundation

- "Step 4: Create and configure an AWS Certificate Manager CA template" on page 109 in Trust Protection Foundation

## Step 1: Verify that your Amazon account is ready to go

### First things first

Make sure that you have the correct permissions specified in your AWS policy and the AWS role that you'll use with CyberArk Trust Protection Foundation.

#### What permissions do I need?

The operations performed by CyberArk's ACM driver depend on a set of permissions found in your AWS policy. The following AWS policy represents the least privilege access required to support public and private certificate management.

You'll need to add additional permissions if you're utilizing other driver features, such as installing certificates into the Amazon IAM certificate store or Elastic Load Balancer.

```
{

        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                        "acm:RequestCertificate",
                        "acm:GetCertificate",
                        "acm:DeleteCertificate",
                        "acm:ImportCertificate",
                        "acm:ListCertificates",
                        "ec2:DescribeInstances"
                        "acm-pca:ListCertificateAuthorities",
                        "acm-pca:GetCertificateAuthorityCertificate",
                        "acm-pca:GetCertificate",
                        "acm-pca:IssueCertificate",
                        "acm-pca:RevokeCertificate",
                ],

                "Resource": "*"
            }
        ]
}
```

You can remove specific features that you don't plan to use by simply omitting lines from the policy. For example, if you know that you will not be using ACM, delete any actions that start with "*acm:*".

For additional information, see "AWS permission requirements" on page 337.

## Step 2: Set up an Amazon Private CA (Optional)

If you're going to use an Amazon Private CA to generate private digital certificates, then you'll need to complete the following tasks in Amazon Certificate Manager. (ACM).

1. Create and configure an Amazon Private CA within a specific region, if you've not already done so.

   Be sure to note the region because you'll need that information when configuring CyberArk's Amazon AWS driver.

2. Update the AWS policy to give permissions to the Private CA (see "Integrating Amazon Certificate Manager with Trust Protection Foundation" on page 104 in the section above).

3. Give the AWS role you'll use with CyberArk's Amazon Credential(a step you'll complete later) permission to the Private CA.

## Step 3: Create an Amazon Credential

In Trust Protection Foundation, create an Amazon credential and specify the AWS role that has permissions to the private CA in ACM. The AWS role used by the Amazon credential in Trust Protection Foundation *must* have access to the Private CA.

### To create an Amazon Credential

1. From the Certificate Manager - Self-Hosted menu bar, click **Inventory** > **Credentials**, and then click **Create a New Credential**.

2. Click the **Credential Type** list and select **Amazon**.

3. Click **Folder** and select the policy folder in which to create your new credential.

4. In **Credential Name**, type a unique name for the new credential object, and then click **Create and Configure**.
5. Click the **Source** list and select *Local*, *EC2 Assigned Role*, or *ADFS*, depending on which authentication method you need.
6. (Conditional) If you selected **Local** from the **Source** list, then do the following:
   a. Type a password in the **Access Key** and **Secret Key** fields, respectively.

      You'll be required to retype them in each of the confirmation fields.

   b. (Optional) If you plan to use this new Amazon Credential to access multiple AWS accounts (using cross-accounts), then in **Role Name** or **Role to Assume**, type just the AWS role name you've set up in AWS (no need to enter the entire ARN).

      If you are using cross-accounts, see Authenticating to multiple AWS accounts using a single Amazon Credential.

   c. In cases where you need an External ID, you can type it here in the **External ID** field.

      ### About Amazon's External ID

      The primary function of *External ID* is to address and prevent the "confused deputy" issue. The External ID makes it less likely that a non-Trust Protection Foundation user can access the other AWS account. Without the External ID protection, any IAM user within the AWS account where the Trust Protection Foundation user resides, would be able to access the other AWS accounts simply by knowing the name of the role in the other account.

      When applying account credentials to provision a certificate, Amazon requires an External ID. The value must have a minimum of 2 characters and a maximum of 1,224 characters. The value must be alphanumeric without white spaces. It can also include the following symbols: plus (+), equal (=), comma (,), period (.), at (@), colon (:), forward slash (/), and hyphen (-).

      For more information, visit Amazon's article, How to Use an External ID When Granting Access to Your AWS Resources to a Third Party.

   d. When you're finished, click **Save**.
7. (Conditional) If you selected **ADFS** from the **Source** drop-down list, then do the following:
   a. In the **ADFS Username Credential** field, select an Amazon credential.

      If you haven't yet created a user name credential for use with Amazon, click **Create New Credential** to define a new one, and then continue.

b. In the **Web Service URL** field, type the full URL of your ADFS server.

c. From the **Role** list, select the account that has the required permissions.

> **DID YOU KNOW?** Each of the roles that appear in the Role list uses the following format:
>
> arn:aws:iam::*AWSAccountNumber*:role/ *RoleMappedToADgroupByADFS*
>
> So, for example:
>
> arn:aws:iam::123423455678:role/MYCO-VenafiTPP
>
> For more information about setting up your federated sign-in through Active Directory (AD) and ADFS, visit https://aws.amazon.com/blogs/security/aws-federated-authentication-with-active-directory-federation-services-ad-fs/.

d. When you're finished, click **Save**.

8. (Conditional) If you selected **EC2 Assigned Role** from the **Source** drop-down list, just click **Save** to finish.

   When you select this option, Trust Protection Foundation authenticates with permissions that are assigned to the EC2 server on which it is running.

   > **IMPORTANT** If you don't see this mode in the Source list, you don't have the proper permissions to use it. You must be either a master administrator, or you must request that you be added to the authorized identities list for using the AWS EC2 Assigned Role. To continue, contact a master administrator. See Authorizing the use of AWS EC2 mode for Amazon Credential.

For additional information, see Creating Amazon credentials in the Administration Guide.

## Step 4: Create and configure an AWS Certificate Manager CA template

To enable Trust Protection Foundation to manage certificates issued by Amazon for use by AWS applications, you must create an AWS Certificate Manager CA template object.

**What's a CA template object? Click Me:**CA templates provide Trust Protection Foundation with the information it needs to request, retrieve and retire certificates issued by the Amazon CA.

### To create and configure an Amazon Certificate Manager CA template

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. From the **Tree** drop-down menu, click **Policy**.
3. In the **Policy** tree, select the folder where you want to create the CA Template object, and then click **Add**.
4. Click **CA Template**, then select **AWS Certificate Manager** to create it.
5. In the **CA Name** box, type a name for the new AWS Certificate Manager object.
6. Under **Configuration**, select the AWS credential you created above (see "Step 3: Create an Amazon Credential" on page 107).
7. From the **Region** list, select the region where your AWS application resides.

   For more information about AWS regions, visit http://docs.aws.amazon.com/general/latest/gr/rande.html.
8. Specify a **Certificate trust type** by selecting either **Public** or **Private certificate**, depending on the type of certificates you want created.
9. (Conditional) If you selected *Public certificate* as your certificate trust type, then do the following:
   a. Click **Validate**.
   b. Under **Options**, select a **Domain Validation Recipient**.

      #### What's this?

      ACM applies standard email-based domain validation requirements before issuing a certificate. Domain validation involves sending an email to a predefined set of administrative recipients (admin, administrator, hostmaster, postmaster, and webmaster) for approval of each of the names in the requested certificate. The email contains a hyperlink that takes the recipient to an approval web page where they click a button to indicate their approval.

ACM allows the certificate requester to choose any sub-domain within the hierarchy of the requested name for the approver. Trust Protection Foundation supports the following recipients. The following examples are applicable for "self.parent.example.com" as the requested name:

- Second-Level Domain (@example.com)
- Parent Domain (@parent.example.com)
- Self Domain (@self.parent.example.com)

If there are multiple names in a certificate request and some of the names are sub-domains of another, approving the highest level domain serves as approval for all of the sub-domains.

10. (Conditional) If you selected *Private certificate* as your certificate trust type, then do the following:
    a. Click **Validate**.
    b. Under **Options**, select which private CA to use.
    c. Select **Private key and CSR are generated by CA** if you want ACM to generate them.

       Clear this box if you want Trust Protection Foundation to generate the private key and CSR.

       > **IMPORTANT** Be sure to verify with Amazon or CyberArk regarding any associated costs for generating private keys and CSRs. Pricing models change periodically and you should be aware of the costs before selecting either option.

    d. (Conditional) If you cleared the *Private key and CSR are generated by CA* checkbox, then you've opted to have CyberArk generate them. In this case, do the following:
       i. Select a **Template** for created the private certificate.
       ii. Select the **Signature Algorithm** to be used when generating the certificate.
       iii. Enter a **Validity Period** (in days).
11. Click **Save**.

> To see additional attributes, review the settings on the **Support** tab.

## Provision a certificate to Amazon Certificate Manager

When another CA supplies a certificate for your ACM, you can set up automatic provisioning. Follow the guidelines in "Getting started: automating certificate enrollment and provisioning" on page 15.

### To provision a certificate to the ACM server

1. In the Policy Tree, add a Device that describes how to provision the certificate to the Amazon Certificate Manager.
2. Add an **Amazon AWS** Application, so the *Amazon App* driver has the necessary credentials and connection to provision.
3. Add a CA template that describes the certificate issuer. Upon expiration, the template ensures that the issuing CA can renew the ACM certificate.
4. In the Certificate **Associations** tab, assign the Device.
5. In the Certificate **Settings** tab, assign the CA template.
6. In the Certificate **Summary** tab, click **Renew Now**.

> **TIP** If you are using the REST API, call Certificates/Request. For more information, see Example 2: Provisioning in the Developer's Guide.

# DigiCert CertCentral

These are the steps required to set up the DigiCert CA template object so that CyberArk Trust Protection Foundation™ can initiate and auto-enroll new or to-be-renewed certificate and key generation requests with the DigiCert CA.

Use this section to configure the DigiCert CA template object so that CyberArk Trust Protection Foundation™ can initiate and auto-enroll new or to-be-renewed certificate and key generation requests with DigiCert CertCentral.

The native DigiCert driver supports DigiCert TLS/SSL and Secure Email (S/MIME) certificate products using the same CertCentral CA template and workflow.

**IMPORTANT** Before you begin, make sure you've reviewed and completed all required prerequisite steps for setting up your CA integration. See "About CA driver prerequisites" on page 17.

This chapter contains the following topics:

## Gathering required information

Use this topic as a worksheet for gathering information you'll be required to use while configuring certificate enrollment using DigiCert.

Before continuing, make sure you have completed the prerequisite steps common to most supported CAs. If not, review "About CA driver prerequisites" on page 17.

For each DigiCert certificate product you issue through Trust Protection Foundation, you must configure a unique CA template that corresponds to a specific DigiCert product.

When you begin configuring Trust Protection Foundation to enroll and issue DigiCert certificates, you'll need the following information:

1. Make sure you've granted 443 network access that is required for Trust Protection Foundation servers configured for certificate management.
2. Get a valid API Key and password credential.
    - Provisioned for API-automated certificate issuance.
3. Indicate each certificate type you intend to manage through CyberArk.
4. Procure an API credential for Trust Protection Foundation.

**NOTE** The list of products available is based on the API key used to authenticate to the DigiCert CA.

| Certificate Type | Will this type be managed? |
|---|---|
| SSL Plus | |
| SSL Multi-domain | |
| SSL Wildcard | |
| SSL EV Plus | |
| SSL EV Multi Domain | |
| Secure Email for Employee | |
| Secure Email for Individual Mailbox | |
| Secure Email for Organization | |

## Secure Email (S/MIME) considerations

If you plan to manage Secure Email (S/MIME) certificates:

- At least one Email Subject Alternative Name (SAN) is required in the certificate request.
- The email address must be in a valid email format.
- Additional certificate request fields appear when a Secure Email product is selected.
- DigiCert may enforce identity validation requirements based on the selected Secure Email product.

Secure Email certificates use the same DigiCert CertCentral CA template configuration workflow as TLS certificates. Field-level configuration details are described in DigiCert CertCentral–certificate settings.

## What's next?

In Trust Protection Foundation, create and configure a DigiCert CA template.

For specific information about DigiCert, see "Configuring the DigiCert CA template object" below.

For general information, see "Step 1: Create CA templates" on page 22 from the section "Getting started: automating certificate enrollment and provisioning" on page 15.

# Configuring the DigiCert CA template object

To enable Trust Protection Foundation to manage DigiCert CertCentral certificates, you must use CyberArk's DigiCert CA template to create and configure a DigiCert CA object.

The DigiCert CA object provides Trust Protection Foundation with the information needed to request, retrieve, and install certificates issued by DigiCert CertCentral. This template supports both TLS/SSL and Secure Email (S/MIME) certificate products available in your DigiCert account.

> **NOTE** DigiCert CertCentral does not allow the revocation of a single generation of a certificate. You must revoke the active certificate in order to revoke all generations.

> **BEST PRACTICE** Consider managing CA Template object settings using a policy. For more information, see "Managing CA templates using policies" on page 58.

### To create and configure a DigiCert CA object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. From the **Tree** drop-down menu, click **Policy**.
3. In the **Policy** tree, select the folder where you want to create the CA Template object, and then click **Add**.
4. Click **CA Template**, then select **DigiCert** to create it.
5. In the **CA Name** box, type a name for the new DigiCert object.

6. Refer to the following table to complete the remaining CA template settings:

| Field | Description |
|---|---|
| **Account Settings** | |
| API Key | A key used by the DigiCert CA to identify your organization. This key is encrypted using a password credential.<br><br>When you create a password credential for use with a DigiCert CA template, you must have used the DigiCert API key as the password. If you didn't, you will likely get an error when you save the DigiCert template.<br><br>If you need to learn how to create a password credential, see Creating credential objects. |
| Validate | Click **Validate** to verify the API Key. If an access_denied_invalid_key error occurs, open the Credential, specify the DigiCert password, and then retry.<br><br>Once the credentials are validated, the **Product Name** list (below, under **Options**) is populated with the products available to your organization.<br><br>**NOTE** The templates that appear in the Product Name list depend on how you've set up your DigiCert account. |
| | |
| **Options** | Selects whether the approval process is manual or automatic. |
| Product Name | Select from the list of products available based on the API Key used to authenticate to the DigiCert CA.<br><br>The DigiCert driver supports both TLS/SSL and Secure Email (S/MIME) products, including:<br><br>◦ Secure Email for Employee<br><br>◦ Secure Email for Individual Mailbox<br><br>◦ Secure Email for Organization |

| Field | Description |
|---|---|
| | When a Secure Email product is selected, additional certificate request fields appear when configuring certificate objects. These fields are described in "DigiCert CertCentral–certificate settings" on page 123. |
| Organization | Select the organization licensed to use the DigiCert product.<br><br>Generally, this value should be the same as the Organization value specified in the CSR. It does not have to be identical because DigiCert uses *fuzzy* matching logic to allow for minor differences. |
| Subject Alt Name Enabled | Configures the current CA template object to support CSRs with DNS-based subject alt name (SAN) values.<br><br>**NOTE** DNS SANs are included in the Unified Communications Certificates that are used with Microsoft Exchange 2007 and Microsoft Office Communications Server.<br><br>If you do not select this option, the current CA template object will not accept CSRs with SAN values. If Trust Protection Foundation attempts to submit a CSR with SAN values, the CA Template object returns the following error:<br><br>Stage 400 (Creating CSR) SubjectAltName not supported by Application and CA.<br><br>**IMPORTANT** Before selecting this option, you must also verify the SAN feature is enabled on your CA engine. It it isn't, the CA returns the same error message when Trust Protection Foundation attempts to submit a CSR with SAN values.<br><br>For Secure Email (S/MIME) products, at least one Email SAN is required in the certificate request. |

| Field | Description |
| --- | --- |
| Signature Algorithm | Lets you select the algorithm DigiCert uses when signing a requested certificate.<br>You must select one of the following supported signature algorithms:<br>◦ SHA256 (default)<br>◦ SHA384<br>◦ SHA512 |
| Organizational Unit Override | (Optional) Type one or more OU values to submit to DigiCert with the CSR to override the OUs specified by policy or in the CSR. You must enter one OU value per line.<br>If no organizational unit override values are specified on the CA template, then the OUs specified on the CSR are submitted to DigiCert with the CSR. |
| Allow Reissuance | Check the **Allow Reissuance** box and type a number (in days) in the **Renewal Window** field if you want to allow some certificate renewals to be handled as *reissuance* requests. This setting is disabled (unchecked) by default.<br>Allow Reissuance is useful in situations where you need to renew a certificate but do not want to change its original expiration date.<br>If you disable Allow Reissuance and a certificate's remaining validity is outside of the Renewal Window, then an error should result indicating the certificate cannot be reissued because the CA template doesn't allow it. If that occurs, then you'll need to request that a Trust Protection Foundation administrator enable Allow Reissuance on the CA template, revoke and then renew the certificate, or create a new certificate object and enroll it using the same CA template (since it would be treated as a first-time enrollment).<br>**TIP** If you issue a certificate using the Specific End Date option that is set outside of the renewal window, the certificate is reissued with the expiration date you specified in Specific End Date. |

| Field | Description |
| --- | --- |
| | **EXAMPLE** Suppose you're updating SHA1 certificates to SHA2. However, you know that updating them could result in changing their expiration dates to fall inside of a *freeze period*– a time in which your IT department is verifying network and system stability. To avoid this collision, you enable Allow Reissuance to ensure that the certificate's expiration date remains the same. You accept the default 90-day renewal window because it is longer than your freeze period. |
| Renewal Window (days) | Specify the number of days before expiration that you plan to renew certificates. The default renewal window is 32 days for all newly created DigiCert templates.<br><br>Typically, 30 to 90 days ensures sufficient time for all the necessary approvals and processes to be completed so that the renewed certificate is in place before the old certificate expires.<br><br>You must select a value that is greater than 0 and less than (or equal to) 365 days. |
| Manual Approval | Requires manual approval for all CSRs submitted using the current CA Template object.<br><br>When you enable Manual Approval, the administrator must log in to the DigiCert CertCentral CA service and manually approve the renewing certificate. |
| Skip Approval | The DigiCert driver updated to support DigiCert's new *Skip Approval* option for faster performance.<br><br>**DID YOU KNOW?** Skip Approval streamlines the processes for an increase in performance of up 30%. To take advantage of this, the Manual Approval option has been removed from the CyberArk DigiCert driver workflow resulting in quicker certificate approvals. |
| Certificate Transparency | Select **Send certificate to a CT log server** to enable support for your Entrust CT log server. This is the default setting.<br><br>To disable it, select **Do not send certificates to a CT log server**. The second option requires that you have configured your DigiCert account to allow this action. |

| Field | Description |
| --- | --- |
| | **To configure DigiCert CertCentral to allow disabling CT log server function**<br>a. Open the DigiCert CertCentral web portal.<br>b. Click **Settings** > **Preferences**.<br>c. Click **Advanced Settings**, and then scroll down to the **Certificate Requests** section.<br>d. Select the **Allow users to change CT logging per request** check box to opt out of CT logging for certificates enrolled using a CA template. |
| Validity Period | Assign a validity period to be used when enrolling certificates. Options include 90 days, 6 months, and 1 year (default).<br>**TIP** If you need to define multiple validity periods, create additional CA objects that differ only in the validity period assignment. |

| Field | Description |
|---|---|
| Allow Users to Specify End Date | The Allow Users to Specify End Date option lets users specify expiration (end) dates for certificates requested from the CA so that they do not expire during your known freeze periods.<br><br>Typically, renewing certificates that expire during freeze periods requires a more challenging approval process. Setting expiration (or end) dates that fall outside of freeze periods avoids potential interruptions.<br><br>When a specific end date has been specified, the issued certificate has that date as its expiration date.<br><br>This check box is cleared after successful enrollment (so that the validity period takes effect thereafter). |
| Custom Fields | Custom Fields allow you to extend out-of-the-box capabilities to capture data points unique to your environment.<br><br>Before custom fields can be selected on a CA template or certificate object, they must first be created in Aperture (**Configuration > Custom Fields**) and applied to the Certificate object type. For more information on creating or configuring custom fields, see Working with custom fields.<br><br>**To enable Custom Fields with DigiCert CertCentral:**<br><br>a. Click **Add/Remove** below Custom Fields.<br>The **Add or Remove Custom Fields** dialog box appears.<br><br>b. Select a previously created custom field in the **Select Custom Field** drop down.<br><br>c. Click **Add**.<br>Your previously selected custom field add to the **Custom Field(s)** box.<br><br>d. Click **Done**.<br><br>**NOTE:** Any custom field configured here that is not also present in your DigiCert account will be silently ignored during the enrollment process. Ensure your DigiCert account is configured to accept these specific fields to avoid data loss. |

7. (Optional) To see additional attributes, review the settings on the **Support** tab.

8. Click **Save**.

## What's next?

After you create a CA object, you can select it from the Policy tree, and then view important information and manage various settings.

- Click the **General** tab to view and modify log and permissions settings.
  - Click the **Log** sub-tab to view any logged events that are triggered by the template object.

    **IMPORTANT** You must have the *Read* permission to view the Log tab.

    For more information about options found on the Log tab, see Viewing log events.
  - On the **Permissions** sub-tab, you can configure the users or groups to whom you want to grant permissions to the new template object.

    Consider managing object permissions via parent objects so that you can take advantage of inheritance. For more information, see Permission inheritance and flow down.

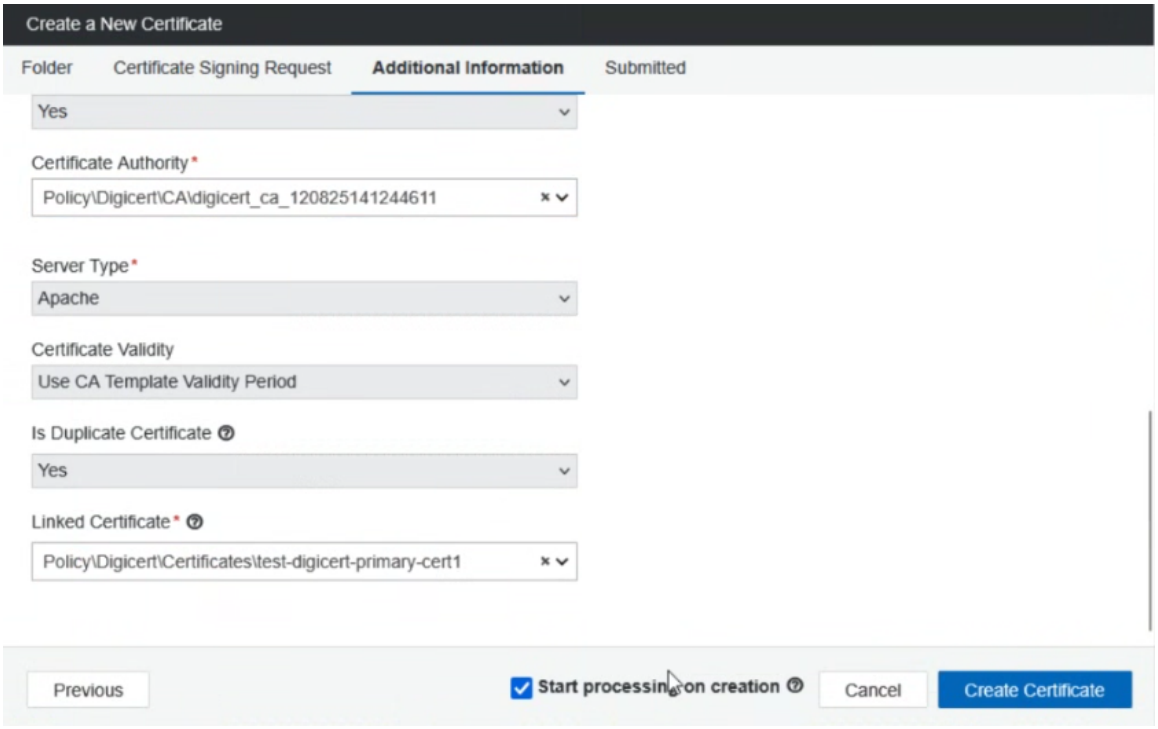
## DigiCert CertCentral–certificate settings

When you define a CA template object, you provide the information Trust Protection Foundation needs to connect to the CA. However, when you associate the DigiCert CA object with a specific certificate object, you must define additional settings that are specific to the certificate–information such as the Validity Period, Server Count, and Server Type. These settings are passed to DigiCert CertCentral when Trust Protection Foundation renews the certificate.

When you select the CA template in the certificate object, you must complete the DigiCert settings.

If the associated DigiCert CA template uses a Secure Email (S/MIME) product, additional certificate-specific fields appear on this page. These fields apply only to Secure Email certificate requests.

In the Policy Tree, the DigiCert CA settings are presented on the certificate object configuration page when you select the DigiCert CA Template object.





The following table describes the certificate object settings.

| Field | Policy | Description |
|---|---|---|
| **Settings** | | |
| Server Type | Yes | Type of server that the certificate is installed on. |
| Certificate Validity | No | Specify the period of time that the certificate is valid by selecting one of the following options:<br>▪ **Use CA Template Validity** |

| Field | Policy | Description |
|---|---|---|
| **TIP** The Certificate Validity option is available only when the **Allow Users to Specify End Date** option is enabled on the associated DigiCert CA template. For more information, see "Allow Users to Specify End Date " on page 122. | | **Period**: Select this option of you want to use the validity period specified on the associated DigiCert CA template. For more information, see "Validity Period" on page 121.<br>■ **Specify End Date**: Select this option and then type an expiration date in the **End Date** field for certificates requested from the CA so that they don't expire during your known freeze periods.<br>If a specific certificate is being renewed or replaced, the Specify End Date is ignored. If a specific end date is required for a certificate that needs to be renewed or replaced, then create and enroll it using a new certificate object. |
| Duplicate Certificate | Yes | ■ Allows you to configure linked certificate information when requesting a certificate<br>■ Enabling this feature will allow you to specify where that linked certificate to act as primary. |
| Linked Certificate | No | Specifies the primary certificate object used as the parent for a duplicate request. This allows the CA to issue child or duplicate certificates linked to the original order ID. |

| Field | Policy | Description |
|---|---|---|
| | | **TIP** The Linked Certificate setting is only available if a duplicate certificate is enabled.<br><br>A linked certificate must be a pre-enrolled certificate. |
| **S/MIME Settings** | | |
| Common Name Indicator | Yes | Determines how the certificate Common Name (CN) is constructed for Secure Email (S/MIME) certificates.<br><br>Options include Email Address, Given Name + Surname, Pseudonym, and Organization Name.<br><br>This field appears only when the associated CA template uses a Secure Email product. |
| First Name | Yes (Conditional) | Required only when Common Name Indicator is set to "Given Name + Surname".<br><br>Must correspond to the local part of the email address.<br><br>Appears only for Secure Email products. |

| Field | Policy | Description |
|---|---|---|
| Last Name | Yes (Conditional) | Required only when Common Name Indicator is set to "Given Name + Surname".<br><br>Appears only for Secure Email products. |
| Primary Usage | Yes | Specifies how the Secure Email certificate is used.<br><br>Options include Dual Use, Signing Only, and Encryption Only.<br><br>Appears only for Secure Email products. |
| Email SAN | Yes | At least one Email Subject Alternative Name (SAN) must be included for Secure Email certificates.<br><br>If no Email SAN is specified, enrollment fails. |

# Entrust CA Gateway

Enterprise PKIs are comprised of several individual CAs used to enroll certificates. This reality often creates administrative challenges. Entrust CA Gateway offers Entrust customers with a single API endpoint for requesting certificates from their many Entrust Security Manager CAs.

You can use CyberArk's Entrust CA Gateway driver to request server, device, user, and code signing certificates issued by Entrust Security Manager CAs.

When you're ready to get started, see the next topic.

## Setting up Entrust CA Gateway

Entrust knows their stuff best, so we won't presume to give you all the details when using Entrust products. We strongly suggest that you carefully review and follow Entrust documentation.

At a minimum, make sure that you've granted the necessary permissions on Entrust to the account that'll be used by Trust Protection Foundation.

When you've done the Entrust thing, continue to the next topic to Configure Trust Protection Foundation for connecting with Entrust CA Gateway.

## Setting up CyberArk for integration with Entrust CA Gateway

Make sure you've already configured Entrust before continuing. Your Entrust service account must be operational for CyberArk to connect and do its thing.

### First things first

Let's take care of a couple of prerequisites first.

- Verify network connectivity to Entrust CA Gateway.
- Confirming that the CA's web service is reachable helps to ensure that the Trust Protection Foundation server can also reach that service. Use whatever tool or method you want to use to send a simple request to your CA's web service. For example, you could use the Windows command line utility and send a ping request.
- Update your firewall rules to allow the Trust Protection Foundation server to connect to the CA web service.
- Make sure you have the Create permission to the Policy folders where you'll be creating credentials and CA template objects.

  > **TIP** Not sure how to set permissions on Policy folders? See Setting policy on a folder (Aperture), and then come back when you're ready.

- Create a certificate credential and import the certificate and private key used to authenticate with Entrust CA Gateway. We'll show you how to do that next.
- Have your Entrust Web Service URL handy.

  Example: https://webservices.host.com:1234/cagw

### Creating a certificate credential

First, you need to create a certificate credential where you'll import the certificate and private key used to authenticate with Entrust CA Gateway.

### To create the new certificate credential

1. From the Certificate Manager - Self-Hosted menu bar, click **Inventory** > **Credentials**.
2. Click **Create a New Credential**.
3. Click **Credential Type** and select **Certificate**.
4. Click **Folder** and select the policy folder in which to create your new credential.
5. In **Credential Name**, type a unique name for the new credential object, and then click **Create and Configure**.
6. In **Edit Credential Settings**, click the **Certificate** field to locate and select the certificate to use for this credential.

   The certificate you select must have a corresponding private key stored in **Trust Protection Foundation**.
7. When you're finished, click **Save**.

## Create and configure an Entrust CA Gateway object

A CA template object provides the information Trust Protection Foundation needs to request, renew, and revoke certificates while also enabling automated provisioning of the certificate to associated devices.

### To create and configure Entrust CA Gateway object

1. From the Certificate Manager - Self-Hosted menu bar, click **Policy Tree**.
2. From the **Policy** tree, select the folder where you want to create the **Entrust CA Gateway** object, and then click **Add** > **CA Template** > **Entrust CA Gateway**.
3. In **CA Name**, type a name for the new template.
4. (Optional) Do the following:
   a. If you find it helpful, type a description for the new template. This might help other users who might want to work with your template.

   b. Click and select other users who should receive email notifications related to your new Entrust CA Gateway.

If another system administrator manages this new CA, it can be helpful to add them as contacts so they'll be notified about changes to this object's settings when they occur.

5. Under **Connection**, do the following:
   a. Enter the Entrust Web Service URL. Example: https://webservices.host.com:1234/cagw
   b. Click and select the certificate credential you created previously, and then click **Validate**.
   c. Trust Protection Foundation parses the credential to ensure it's valid, which occurs when CA and Profile under Options are populated.
6. Under **Options**, do the following:
   a. Select one of the available CAs from the CA drop-down.
   b. Select one of the available profiles from the Profile list.
   c. Depending on which profile you've selected, do the following:
      i. (Optional) When **Keys are generated by the CA** is selected, the format of the certificate request is PKCS#12.
      ii. (Optional) Select **Exclude subjectVariables when requesting certificates**, if you do not want the subject variables (CN, O, OU, L, ST, C, and DC) included in the certificate request.
      iii. (Optional) Select **Subject Alt Name Enabled** if you want this CA object to support CSRs with DNS-based subject alt name (SAN) values.

      If you do not select this option, the current CA template object will not accept CSRs with SAN values. If Trust Protection Foundation attempts to submit a CSR with SAN values, the CA Template object returns the following error:

      ```
      Stage 400 (Creating CSR) SubjectAltName not supported by Application and CA.
      ```

      For additional information on defining SAN values in a certificate, see Subject Alt Name in the topic About certificate object settings (Policy Tree).
      iv. (Optional) Select **Allow Users to Specify End Date** if you want this CA object to let users specify expiration (end) dates for certificates requested from the CA.

      The Allow Users to Specify End Date option lets users specify expiration (end) dates for certificates requested from the CA so that they do not expire during your known freeze periods.

Typically, renewing certificates that expire during freeze periods requires a more challenging approval process. Setting expiration (or end) dates that fall outside of freeze periods avoids potential interruptions.

When a specific end date has been specified, the issued certificate has that date as its expiration date.

This check box is cleared after successful enrollment (so that the validity period takes effect thereafter).

v. In **Validity Period**, type a number (in days) that certificates are valid.

This setting is used when enrolling certificates. You can enter any positive integer. The default is 365 (days).

Assign a validity period to be used when enrolling certificates. Options include 90 days, 6 months, and 1 year (default).

> **TIP** If you need to define multiple validity periods, create additional CA objects that differ only in the validity period assignment.

7. (Optional) To see additional attributes, review the settings on the **Support** tab.

8. When you're finished, click **Save**.

## What's next?

After you create a CA object, you can select it from the Policy tree, and then view important information and manage various settings.

- Click the **General** tab to view and modify log and permissions settings.
  - Click the **Log** sub-tab to view any logged events that are triggered by the template object.

    > **IMPORTANT** You must have the *Read* permission to view the Log tab.

    For more information about options found on the Log tab, see Viewing log events.
  - On the **Permissions** sub-tab, you can configure the users or groups to whom you want to grant permissions to the new template object.

    Consider managing object permissions via parent objects so that you can take advantage of inheritance. For more information, see Permission inheritance and flow down.

# GlobalSign MSSL

The following sections review the steps required to set up the GlobalSign MSSL CA driver.

This chapter contains the following topics:

## Configuring the GlobalSign MSSL CA Template object

To enable Trust Protection Foundation to manage GlobalSign certificates, you must configure the GlobalSign MSSL CA Template object. This object provides the information Trust Protection Foundation needs to request, retrieve, and install certificates issued by the GlobalSign MSSL CA.

> **BEST PRACTICE** Consider managing CA Template object settings using a policy. For more information, see "Managing CA templates using policies" on page 58 in the *Certificate Authority and Hosting Platform Integration Guide*.

### To create and configure a GlobalSign MSSL CA template

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. From the **Tree** drop-down menu, click **Policy**.
3. In the **Policy** tree, select the folder where you want to create the CA Template object, and then click **Add**.
4. Click **CA Template**, then select **GlobalSign MSSL** to create it.
5. In the **CA Name** box, type a name for the new GlobalSign MSSL object.

Refer to the following table to complete the remaining CA template settings:

| Field | Description |
|---|---|
| **Connection** | |
| Profile ID | The profile ID contains a string of numbers and letters that are associated with your vetted certificate profile. The Profile ID can be determined by logging in to the GlobalSign MSSL portal, clicking on the **Manage Domains & Profiles** link, and then clicking on the **Toggle display of Profile ID & Domain ID** link. |
| Domain ID (Optional) | The Domain ID contains a string of numbers and letters that are associated with each domain for which you can issue certificates. |

| Field | Description |
|---|---|
| | The Domain ID can be determined by logging in to the GlobalSign MSSL portal, clicking on the **Manage Domains & Profiles** link, and then clicking on the **Toggle display of Profile ID & Domain ID** link.<br><br>If you choose not to specify a domain ID, then certificates may be enrolled using any domains that have been validated in your GlobalSign account. This allows for individual certificates that reference more than one domain and allows one CA template to be used for enrolling certificates that specify different domains. When the domain ID is specified, only that domain may appear in the certificate request. |
| Credential | Username and password that is used to authenticate to the GlobalSign MSSL portal. |
| Validate | Click this button to validate the Profile ID, Domain ID, and Credential settings by attempting to connect to and authenticate with GlobalSign. Once these credentials are validated, the Template list is populated with the products available to your organization. |
| Connect to Test Service | Configures the CA template to use GlobalSign's non-production service for testing purposes.<br><br>When this option is enabled, certificates enrolled using the CA template are not generally trusted by clients because they are issued by an untrusted certificate authority. |
| **Options** | |
| Template | Select one of the following certificate templates, which are retrieved from GlobalSign for issuance of certificates.<br><br>■ Organization SSL<br><br>■ Extended Validation (EV) SSL<br><br>These templates are supported for domains that have completed the respective level of validation (OV or EV).<br><br>One instance of the CA template must be created for each product template that corresponds to certificates that will be issued to the organization. |

| Field | Description |
|---|---|
| SAN Enabled | Select this option if you want this instance of the CA template to allow certificates to which it has been assigned to use Subject Alternative Names. |
| SAN Type | Identifies the format of Subject Alternative Name that is supported by this instance of the CA template.<br>■ **Fully Qualified Domain**: Allows for a subject alt name such as domain.com to be used as subject alt name in the certificate.<br>■ **Internal hostname or IP Address**: Allows for a subject alt name such as an internal server name (should not end with domain.com) or an IP Address to be used as subject alt name in the certificate.<br>■ **Subdomain**: Allows for a subject alt name such as sub.domain.com to be used as subject alt name in the certificate.<br>■ **Unified Communication Certificate**: Allows for a subject alt name such as 'mail.domain.com' to be used as subject alt name in the certificate. |
| Signature Algorithm | Specifies the algorithm to be used by GlobalSign when it signs a requested certificate. Supported values are SHA1 (default) and SHA2-256. |

| Field | Description |
|---|---|
| **Validity Period** | |
| Available Validity Periods (Months) | Validity Periods that are supported by the GlobalSign MSSL CA. |
| Supported Validity Periods (Months) | Validity Periods that are available to be selected when configuring a GlobalSign MSSL certificate.<br><br>To populate this list, select the Supported Validity Periods that you want to be available when configuring GlobalSign MSSL certificates in Policy Tree, and then click the right-arrow.<br><br>Press **Shift+click** to select multiple, contiguous items.<br><br>Press **Ctrl+click** to select multiple, discontiguous items. |

7. (Optional) To see additional attributes, review the settings on the **Support** tab.
8. Click **Save**.

## What's next?

After you create a CA object, you can select it from the Policy tree, and then view important information and manage various settings.

- Click the **General** tab to view and modify log and permissions settings.
  - Click the **Log** sub-tab to view any logged events that are triggered by the template object.

    **IMPORTANT** You must have the *Read* permission to view the Log tab.

    For more information about options found on the Log tab, see Viewing log events.
  - On the **Permissions** sub-tab, you can configure the users or groups to whom you want to grant permissions to the new template object.

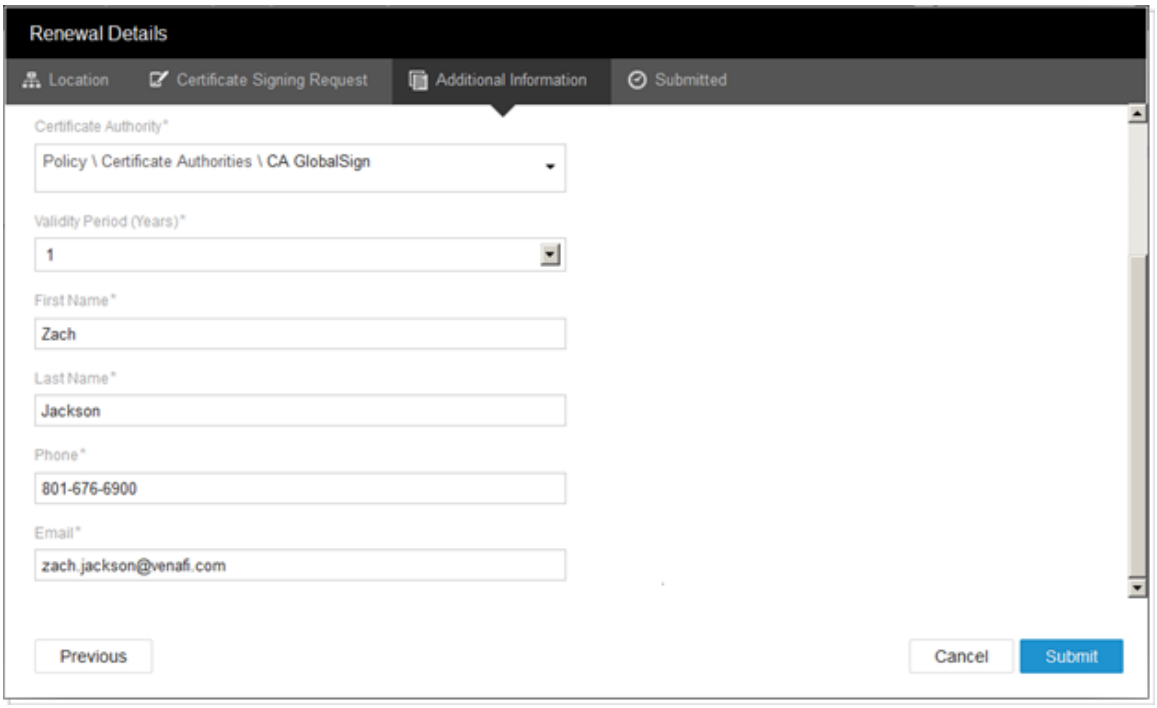
    Consider managing object permissions via parent objects so that you can take advantage of inheritance. For more information, see Permission inheritance and flow down.

## GlobalSign MSSL Settings–certificate settings

When you define a CA Template object, you provide the information Trust Protection Foundation needs to connect to the CA. However, when you associate the GlobalSign MSSL CA Template object configuration with a specific Certificate object, you must define additional settings that are specific to the certificate–information such as the Validity Period and details that identify the Certificate Owner. These settings are passed to the GlobalSign MSSL CA when Trust Protection Foundation renews the certificate.

When you select the CA Template in the Certificate object configuration, you must complete the GlobalSign MSSL settings.

- The GlobalSign MSSL settings appear in the **Additional Information** section of the **Renewal Details** page.




The following table reviews the GlobalSign MSSL settings that apply to certificate objects.

| Field | Policy | Description |
|---|---|---|
| **Settings** | | |
| Validity Period | No | Period of time (in months) that the certificate is valid.<br><br>The options available in this menu are determined by the Available Validity Periods configured in the GlobalSign CA Template object configuration. For more information, see "Validity Period" on page 136. |
| **Certificate Owner** | | |
| First Name |  | GlobalSign certificate owner's first name. |
| Last Name |  | GlobalSign certificate owner's last name. |
| Phone |  | GlobalSign certificate owner's telephone number. |
| Email |  | GlobalSign certificate owner's email address. |

## Google Cloud CA Service

When you use Google Cloud CA Service with CyberArk Trust Protection Foundation, you can simplify your enterprise PKI strategy. There's no need to deploy additional servers; once configuration is complete, simply request a new certificate. As a PKI administrator, you'll have a much simpler way to revoke trust since extracting certificates from certificate authorities (CAs) in your network can be cumbersome. And of course, to keep things secure, you can keep your root certificates in a local CA.

**DID YOU KNOW?** Lots of companies are moving workloads to cloud services. Cloud services offer many advantages over locally hosted services, including

- eliminating the overhead of infrastructure and application maintenance
- globalizing all of your services, and
- higher and cheaper availability

With its cloud-based PKI solution, Google enables customers to replace some or all of their existing PKI infrastructure with Google Cloud-hosted private CAs. This in turn reduces their administrative burden and costs. And as a bonus, they don't have to worry about extending their PKI to cloud services to meet certificate needs of cloud-based workloads.

The relative simplicity of a Cloud solution combined with the power of Trust Protection Foundation makes system outage prevention simpler.

If you're already a pro at Google Cloud CA Service, this should be a snap. If you're new to Google products, you'll need to know how to create certificate authorities, grant permissions, and create roles. For the CyberArk side of the equation, we've got you covered.

This chapter contains the following topics:

## First things first

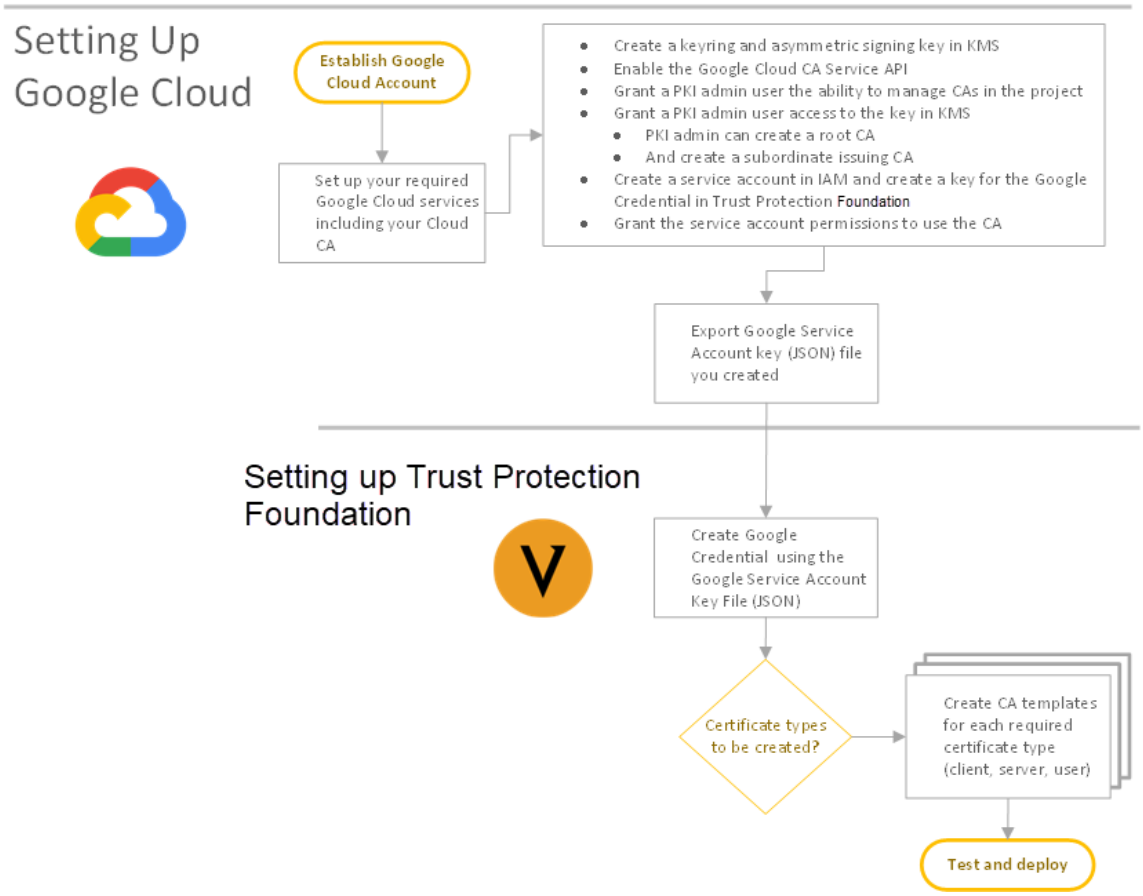
Let's take care of a couple of prerequisites first.

- Make sure you have the correct permissions to the Policy folder in Policy Tree where you'll be doing some configuration stuff. Specifically, you'll need
    - View/Read/Write permissions for uploading your Google Service Account Key File
- Make sure you have your Google Service Account Key File ready for uploading to Trust Protection Foundation. You'll need it!
- If you want this CA object to support CSRs with DNS-based subject alt name (SAN) values, verify that the SAN feature is enabled on your CA engine; you can then enable Select Subject Alt Name when configuring CyberArk.

  For additional information on defining SAN values in a certificate, see Subject Alt Name in the topic About certificate object settings.

> **TIP** Not sure how to set permissions on Policy folders? See Setting policy on a folder (Aperture), and then come back when you're ready.

From a *high level*, here are the steps for integrating Google Cloud CA with Trust Protection Foundation, from setting up your Google account to giving your users access to Google Cloud CA-generated certificates that are secured and managed by Trust Protection Foundation:




## Setting up Google Cloud

Google knows their stuff best, so we won't presume to give you all the details when using Google products. We strongly suggest that you carefully review and follow Google documentation to establish your Google Cloud account, if you haven't already. When you've done that, return here to proceed.

The least permissions required to request, retrieve, and revoke certificates for the service account are

- privateca.locations.get
- privateca.locations.list
- privateca.caPools.get
- privateca.caPools.list

- privateca.certificateAuthorities.get
- privateca.certificateAuthorities.list
- privateca.certificates.get
- privateca.certificates.create
- privateca.certificates.update

After you've completed your Google configuration, see "Setting up CyberArk for use with Google Cloud CA Service" below.

## Setting up CyberArk for use with Google Cloud CA Service

Make sure you've already configured Google before continuing. Your Google account must be up and operational for CyberArk to connect and do its thing.

For Google Cloud Private CA, if you are upgrading from 21.1 or earlier, be sure you have the correct version. Support for Google Cloud Private CA ends sometime in Q2 of 2021. You must upgrade to Version 1.0 or greater. After the Google upgrade:

- You should either re-validate existing CA Templates or create new ones.
- Your existing certificates can not be managed by Google anymore. So, you must use the new or updated CA Template to re-issue certificates.

### Creating a Google Credential

A Google credential object is used by CyberArk Trust Protection Foundation to store your *Google Cloud Service Account Key file* (JSON format). That's so that Trust Protection Foundation can use your Google Service Account to access your Google Cloud CA and Google Cloud Load Balancer.

#### To create a new Google credential

1. From the Platform menu bar, click Policy Tree.
2. From the **Policy tree**, select the folder where you want to create the Google credential, and then click **Add** > **Credential** > **Google Credential**.
3. In **Credential Name**, type a name for the new credential.

4. Click **Upload Google Service Account Key File** (near the top of the page), locate the json file, and then click **Upload**.

5. Click **Save**.

## Create and configure a Google Cloud CA template

To enable Trust Protection Foundation to manage certificates installed anywhere, including on Google Cloud-native applications such as Google Cloud Load Balancer, you must configure a Google Cloud CA application object. This object provides the information Trust Protection Foundation needs to request, renew, and revoke certificates while also enabling automated provisioning of the certificate to associated devices.

### To create and configure a new Google Cloud CA template

1. Log in to Policy Tree, or use the Product Switcher if you're in Aperture.

2. From the **Policy tree**, select the folder where you want to create the Google Cloud CA template, click **Add** > **CA Template** > **Google Cloud CA**.

   > **TIP** Consider these concepts about organizing folders in the Policy tree to maximize ease of use.

3. In **CA Name**, type a name for the new template.

4. (Optional) If you find it helpful, type a **Description** for the new template.

   This might help other users who might want to work with your template.

5. (Optional) In **Contacts**, click  and select other users who should receive email notifications related to your Google Cloud CA.

   > **TIP** If another system administrator manages this new CA, it can be helpful to add them as contacts so they'll get notified about changes to this object's settings, when they occur.

6. In the **Connection** box, click  and select the Google credential you created previously.

7. The **Project ID** is populated using the project from the valid Google Credential. This provides a way to specify in which project the Private CA is placed. The value of the field is extracted from the Google Credential, which the user has the option to edit.

8. Click **Validate**.

Trust Protection Foundation retrieves all regions (locations) and all CAs for each of them and then selects the first region that has at least one CA. Certificate Authority ID is populated with the CAs for the selected region. If a region list can't be retrieved, Trust Protection Foundation defaults to the current region.

For Google Cloud Private CA, you can see the **Certificate Authority ID** with additional pool information.

> **TIP** Trust Protection Foundation remembers the last saved ID; so you only need to validate again if you need to update the CA template to use a different ID.

9. Under **Configuration**, do the following:
   a. Select the **Region** where your Google Cloud CA application resides.

   b. (Optional) Select the **Certificate Authority ID** and corresponding CA Pool information. If there are no values, select another **Region** that includes a CA.

   c. (Optional) Select **Subject Alt Name Enabled** if you want this CA object to support CSRs with DNS-based subject alt name (SAN) values.

      If you do not select this option, the current CA template *object will not accept CSRs with SAN values*. If Trust Protection Foundation attempts to submit a CSR with SAN values, the CA Template object returns the following error:

      ```
      Stage 400 (Creating CSR) SubjectAltName not supported by Application and CA.
      ```

      For additional information on defining SAN values in a certificate, see Subject Alt Name in About certificate object settings (Policy Tree).

   d. (Optional) Select **Allow Users to Specify End Date** if you want this CA object to allow users to specify expiration (end) dates for certificates requested from the CA so that they do not expire during your known freeze periods.

      Renewing certificates that expire during freeze periods can be challenging. But you can avoid potential interruptions by setting expiration dates that fall outside of the freeze period.

      > **DID YOU KNOW?** This option is disabled automatically after successful enrollment so that the validity period takes effect thereafter.

e. In **Validity Period**, type a number (in days) that certificates are valid. This setting is used when enrolling certificates. You can enter any positive integer. The default is 365 (days).

> **TIP** If you need to define multiple validity periods, create additional CA objects that differ only in the validity period assignment.

10. (Optional) To see additional attributes, review the settings on the **Support** tab.

11. When you're finished, click **Save**.

### What's next?

After you create a CA object, you can select it from the Policy tree, and then view important information and manage various settings.

- Click the **General** tab to view and modify log and permissions settings.
  - Click the **Log** sub-tab to view any logged events that are triggered by the template object.

    > **IMPORTANT** You must have the *Read* permission to view the Log tab.

    For more information about options found on the Log tab, see Viewing log events.
  - On the **Permissions** sub-tab, you can configure the users or groups to whom you want to grant permissions to the new template object.

    Consider managing object permissions via parent objects so that you can take advantage of inheritance. For more information, see Permission inheritance and flow down.

## HID PKIaaS

In order to use the HID PKIaaS CA driver, your account needs to be prepared by HID PKIaaS for automated access.

### To prepare your account for automated access

1. Obtain a Web Service Signing Certificate (Web API Certificate) issued by HID PKIaaS.

   This certificate is used by the driver to access the HID PKIaaS API.

2. Provide HID PKIaaS with the IP addresses of those Trust Protection Foundation servers that will be connecting to HID PKIaaS's APIs.

   For security reasons HID PKIaaS blocks API access from any address that is not registered. If NAT services are used on your network, you must register your external address.

3. Associate your Web Service Signing Certificate in HID PKIaaS with your IP address, and then configure firewall and web service access.

4. Using the HID PKIaaS web service, select which certificate templates can be requested and configure HID PKIaaS to enable these, accordingly.

   To protect your organization, only templates that you specifically request to be enabled can be used via the web service.

5. Make sure that a subscriber is registered to the account and assigned to the organizations that are going to request certificates through the web service.

The following topics include the steps required to set up the HID PKIaaS CA.

This chapter contains the following topics:

## Configuring the HID PKIaaS CA Template object

To enable Trust Protection Foundation to manage HID PKIaaS certificates, you must configure the HID PKIaaS CA Template object. This object provides the information that Trust Protection Foundation needs to request, retrieve, renew and revoke certificates issued by the HID PKIaaS CA.

**BEST PRACTICE** Consider managing CA Template object settings using a policy. For more information, see "Managing CA templates using policies" on page 58 in the *Certificate Authority and Hosting Platform Integration Guide*.

### To create and configure a HID PKIaaS CA template

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. From the **Tree** drop-down menu, click **Policy**.
3. In the **Policy** tree, select the folder where you want to create the CA Template object, and then click **Add**.
4. Click **CA Template**, then select **HID PKIaaS** to create it.
5. In the **CA Name** box, type a name for the new HID PKIaaS object.
6. (Optional) To see additional attributes, review the settings on the **Support** tab.
7. Click **Save**.

Refer to the following table to complete the remaining CA template settings:

### HID PKIaaS CA Template Configuration

| Field | Description |
|---|---|
| **Connection** | |
| Web API Certificate | The certificate credential issued by the QuoVadis CA, which authenticates access to interact with the HID PKIaaS web service API. |

| Field | Description |
|---|---|
| Subscriber Email | The RFC822 formatted email address of the subscriber account registered with HID PKIaaS and that will be used to request certificates. |
| Account Name | The name of the customer account that is established and provided by HID PKIaaS. |
| Account Organization | The name of the customer organization (or department) that has been approved for enrollment by HID PKIaaS. |
| Subject Alt Name Enabled | Configures the current CA template object to support CSRs with DNS-based subject alt name (SAN) values.<br><br>**NOTE** DNS SANs are included in the Unified Communications Certificates that are used with Microsoft Exchange 2007 and Microsoft Office Communications Server.<br><br>If you do not select this option, the current CA template object will not accept CSRs with SAN values. If Trust Protection Foundation attempts to submit a CSR with SAN values, the CA Template object returns the following error:<br><br>Stage 400 (Creating CSR) SubjectAltName not supported by Application and CA.<br><br>**IMPORTANT** Before selecting this option, you must also verify the SAN feature is enabled on your CA engine. It it isn't, the CA returns the same error message when Trust Protection Foundation attempts to submit a CSR with SAN values.<br><br>For Secure Email (S/MIME) products, at least one Email SAN is required in the certificate request.<br><br>For additional information on defining SAN values in the certificate, see Subject Alt Name in the topic Certificate settings. |
| Use test account | |

| Field | Description |
|---|---|
| Validate | Click this button to validate the connection to the HID PKIaaS CA and to retrieve the Policy Templates that are available for enrollment. |
| **Options** | |
| Template | The Policy Template for which certificates will be requested during enrollment. Policy Templates define Validity Period, Subject DN behavior, support for Subject Alternative Names (SANs), and the certificate type (e.g. Business SSL, EV SSL, etc.). |

## What's next?

After you create a CA object, you can select it from the Policy tree, and then view important information and manage various settings.

- Click the **General** tab to view and modify log and permissions settings.
  - Click the **Log** sub-tab to view any logged events that are triggered by the template object.

    **IMPORTANT** You must have the *Read* permission to view the Log tab.

    For more information about options found on the Log tab, see Viewing log events.
  - On the **Permissions** sub-tab, you can configure the users or groups to whom you want to grant permissions to the new template object.

    Consider managing object permissions via parent objects so that you can take advantage of inheritance. For more information, see Permission inheritance and flow down.

The following table describes settings for the HID PKIaaS CA Template object.

### HID PKIaaS CA Template Object Configuration

| Field | Description |
|---|---|
| **Connection** | |
| Web API Certificate | The certificate credential issued by the QuoVadis CA, which authenticates access to interact with the HID PKIaaS web service API. |
| Subscriber Email | The RFC822 formatted email address of the subscriber account registered with HID PKIaaS and that will be used to request certificates. |
| Account Name | The name of the customer account that is established and provided by HID PKIaaS. |
| Account Organization | The name of the customer organization (or department) that has been approved for enrollment by HID PKIaaS. |
| Subject Alt Name Enabled | Configures the current CA template object to support CSRs with DNS-based subject alt name (SAN) values.<br><br>**NOTE** DNS SANs are included in the Unified Communications Certificates that are used with Microsoft Exchange 2007 and Microsoft Office Communications Server.<br><br>If you do not select this option, the current CA template object will not accept CSRs with SAN values. If Trust Protection Foundation attempts to submit a CSR with SAN values, the CA Template object returns the following error:<br><br>Stage 400 (Creating CSR) SubjectAltName not supported by Application and CA.<br><br>**IMPORTANT** Before selecting this option, you must also verify the SAN feature is enabled on your CA engine. It it isn't, the CA returns the same error message when Trust Protection Foundation attempts to submit a CSR with SAN values.<br><br>For Secure Email (S/MIME) products, at least one Email SAN is required in the certificate request.<br><br>For additional information on defining SAN values in the certificate, see Subject Alt Name in the topic Certificate settings. |

| Field | Description |
|---|---|
| Use test account | |
| Validate | Click this button to validate the connection to the HID PKIaaS CA and to retrieve the Policy Templates that are available for enrollment. |
| **Options** | |
| Template | The Policy Template for which certificates will be requested during enrollment. Policy Templates define Validity Period, Subject DN behavior, support for Subject Alternative Names (SANs), and the certificate type (e.g. Business SSL, EV SSL, etc.). |

### HID PKIaaS Settings—certificate settings

When you define a CA Template object, you provide the information that Trust Protection Foundation needs to connect to the CA as well as several other properties of the certificate to be enrolled. The HID PKIaaS CA template currently does not offer any settings that are specific to the certificate object to which it has been applied.

## Microsoft Active Directory Certificate Services (ADCS) - Enterprise and Standalone

The following section reviews the steps required to set up the Microsoft Active Directory Certificate Services (ADCS) - Enterprise and Standalone CA Template object so that CyberArk Trust Protection Foundation™ can initiate and auto-enroll new or to-be-renewed certificate and key generation requests with the Microsoft CA driver.

Trust Protection Foundation supports all CAs included with supported Windows Server operating systems. For information on the supported Windows Server operating systems, see "Supported integrations with Trust Protection Foundation" on page 18.

**IMPORTANT** To use a Microsoft Active Directory Certificate Services (ADCS) - Enterprise and Standalone CA, you must install Microsoft CA Support and configure the user account required to access the Microsoft Active Directory Certificate Services (ADCS) - Enterprise and Standalone CA.

The user account used to access the Microsoft CA must have *Read*, *Issue and Manage Certificates*, and *Request Certificates* permissions to the Microsoft CA server.

This chapter contains the following topics:

## Configuring the Microsoft template object

To enable Trust Protection Foundation to manage Microsoft certificates, you must configure the Microsoft CA template object. This object provides the information Trust Protection Foundation needs to request, retrieve, and install certificates issued by the Microsoft CA driver. These instructions apply to Microsoft Active Directory Certificate Services (ADCS) - Enterprise and Standalone CA.

**IMPORTANT** Trust Protection Foundation can issue and renew certificates based on Enterprise Microsoft CA templates that build the Subject Name from the request. However, Enterprise Microsoft CA templates that build the Subject Name from Active Directory are *not* supported.

**BEST PRACTICE** Consider managing CA Template object settings using a policy. For more information, see "Managing CA templates using policies" on page 58 in the *Certificate Authority and Hosting Platform Integration Guide*.

### To create a Microsoft CA template object

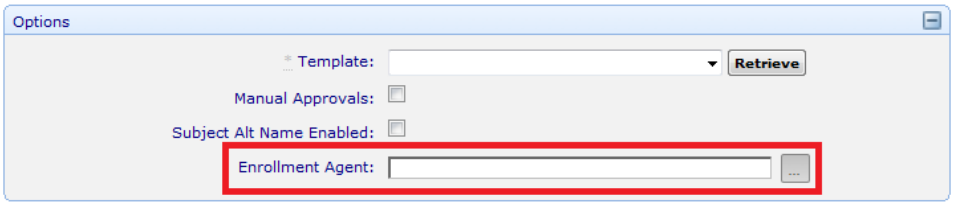
1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. From the **Tree** drop-down menu, click **Policy**.
3. In the **Policy** tree, select the folder where you want to create the CA Template object, and then click **Add**.
4. Click **CA Template**, then select **Microsoft** to create it.
5. In the **CA Name** box, type a name for the new Microsoft object.
6. Refer to the following table to complete the remaining CA template settings:

#### Microsoft CA Template Settings

| Field | Description |
|---|---|
| **Connection** | |
| Hostname | IP Address or DNS name of the Microsoft CA server.<br>Trust Protection Foundation supports both IPv4 or IPv6 connections. |

| Field | Description |
|---|---|
| Service Name | Name of the Microsoft CA service. This will match the Common Name (CN) of the CA's certificate. |
| Credential | User name credential required to connect to the Microsoft CA.<br><br>This account must have Read, Issue and Manage Certificates, Request Certificates permission on the CA. If this is an Enterprise CA, the account also requires Read and Enroll permissions applied to any templates that Certificate Manager - Self-Hosted will use.<br><br>The credential can be expressed in any of the following formats, although the format you use depends upon your Active Directory environment:<br><br>◦ SAM account name<br><br>◦ Domain name syntax (domain_name\username)<br><br>◦ User Principle Name syntax (valid_user@orgUnit.domain.com)<br><br>**To select a credential**<br>a. Click the **Browse** button.<br><br>The **Credential Selector** dialog appears.<br><br>b. Select the user name credential that stores the user name and password required to connect to the Microsoft CA server, and then click **Select**. |
| **Options** | |
| Template | The Microsoft CA template that you want to associate with the current CA Template object. Trust Protection Foundation references this template when it submits the CSR. The template menu is automatically populated when you click **Retrieve**.<br><br>If new templates are added to the CA, you must click **Retrieve** to update the list of available templates.<br><br>The template referenced by Trust Protection Foundation's Microsoft CA template object must be configured to have its Subject Name supplied in the request. |

| Field | Description |
|---|---|
| Allow Users to Specify End Date | Lets users specify expiration (end) dates for certificates requested from the CA so that they do not expire during your known freeze periods.<br><br>Typically, renewing certificates that expire during freeze periods requires a more challenging approval process. Setting expiration (or end) dates that fall outside of freeze periods avoids potential interruptions.<br><br>When a specific end date has been specified, the issued certificate has that date as its expiration date.<br><br>This check box is cleared after a successful enrollment so that the validity period takes effect thereafter.<br><br>**TIP** To use the Allow Users to Specify End Date feature, you *might* need to enable the EDITF_ATTRIBUTEENDDATE flag on the CA, which is typically done using the **certutil** command-line tool:<br><br>`certutil -setreg policy\EditFlags +EDITF_ATTRIBUTEENDDATE`<br><br>You would then restart the **certsvc** service.<br><br>For more information, refer to the following Microsoft articles:<br><br>◦ https://support.microsoft.com/en-us/kb/254632<br><br>◦ https://msdn.microsoft.com/en-us/library/cc226763.aspx<br><br>You can verify Microsoft CA settings by entering the following at a command prompt:<br><br>`certutil -getreg policy\EditFlags` |
| Manual Approvals | Requires manual approval for all CSRs submitted using the current CA Template object.<br><br>When you enable **Manual Approval**, the administrator must log in to the Microsoft CA service and manually approve the renewing certificate. |

| Field | Description |
|---|---|
| Subject Alt Name Enabled | Configures the current CA template object to support CSRs with DNS-based subject alt name (SAN) values.<br><br>**NOTE** DNS SANs are included in the Unified Communications Certificates that are used with Microsoft Exchange 2007 and Microsoft Office Communications Server.<br><br>If you do not select this option, the current CA template object will not accept CSRs with SAN values. If Trust Protection Foundation attempts to submit a CSR with SAN values, the CA Template object returns the following error:<br><br>`Stage 400 (Creating CSR) SubjectAltName not supported by Application and CA.`<br><br>**IMPORTANT** Before selecting this option, you must also verify the SAN feature is enabled on your CA engine. It it isn't, the CA returns the same error message when Trust Protection Foundation attempts to submit a CSR with SAN values.<br><br>For Secure Email (S/MIME) products, at least one Email SAN is required in the certificate request.<br><br>For additional information on defining SAN values in the certificate, see Subject Alt Name in the topic Certificate settings. |
| Automatically include CN as DNS SAN | Makes it so that every certificate enrolled using the CA template will have the its Common Name included as a DNS Subject Alternative Name (SAN).<br><br>As of v1.1.7 of the CA/Browser Forum Baseline Requirements, usage of the Common Name is considered deprecated and to be represented instead by an equivalent DNS SAN. |
| Enrollment Agent | Lets you assign a certificate credential containing an Enrollment Agent certificate for signing requests prior to submitting them to the Microsoft CA.<br><br>  |

| Field | Description |
| --- | --- |
| | For more information, see Establish Restricted Enrollment Agents in the Microsoft TechNet Library. |

7. (Optional) To see additional attributes, review the settings on the **Support** tab.
8. Click **Save**.

## What's next?

After you create a CA object, you can select it from the Policy tree, and then view important information and manage various settings.

- Click the **General** tab to view and modify log and permissions settings.
  - Click the **Log** sub-tab to view any logged events that are triggered by the template object.

    **IMPORTANT** You must have the *Read* permission to view the Log tab.

    For more information about options found on the Log tab, see Viewing log events.
  - On the **Permissions** sub-tab, you can configure the users or groups to whom you want to grant permissions to the new template object.

    Consider managing object permissions via parent objects so that you can take advantage of inheritance. For more information, see Permission inheritance and flow down.

# Setting up redundancy using the Microsoft CA Pool template

If you are using two Microsoft CAs, you can configure a Microsoft CA Pool template to automatically enroll certificates using a second CA in the event that the first CA is not successful. There is no functionality today built into Active Directory Certificate Services (ADCS) that provides redundancy in the event of service

degradation.

> **DID YOU KNOW?** The Microsoft CA Pool is a special type of CA template. Whereas all other CyberArk CA templates define *how Trust Protection Foundation interacts with a specific CA and what type of certificates are issued*, the Microsoft CA Pool does neither, but is instead an aggregation of multiple MSCA templates.
>
> So what happens if the first CA fails? In the event that the first Microsoft CA in the CA pool is not available, Trust Protection Foundation stops querying that CA for 60 minutes and automatically queries the second CA in the CA pool. This process is repeated up to the optional third Microsoft CA.

You can configure a Microsoft CA Pool template in which where you can specify up to three separate MSCA templates.

> **TIP** Before you create your Microsoft CA Pool template, you should already have at least two MSCA templates defined because you'll need to select them as part of this procedure.

### To create a Microsoft CA Pool template object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. From the **Tree** drop-down menu, click **Policy**.
3. In the **Policy** tree, select the folder where you want to create the CA Template object, and then click **Add**.
4. Click **CA Template**, then select **fj** to create it.
5. In the **CA Name** box, type a name for the new Microsoft CA Pool object.
6. Under **Options**,click  to select at least two existing MSCA templates.

   > **BEST PRACTICE** Although not required, all of the MSCA templates assigned to the Microsoft CA Pool should issue certificates that chain up to the same root certificate authority (e.g., they are issuing CAs in the same PKI hierarchy).

7. Select the **Manual Approvals** and **Allows Users to Specify End Date** check boxes to enable them.

> **IMPORTANT** Whether you leave these settings disabled (which is the default setting), or whether you enable them, note that how you set them here overrides the same settings found on the individual MSCA templates you assign to the pool.
>
> For example, if you have *enabled* Manual Approvals on any of the MSCA templates you include in the pool but then *disable* Manual Approvals here in the Microsoft CA Pool settings, Manual Approvals will be disabled when that template is used by your Microsoft CA Pool.

8. When you're finished, click **Save**.

### What's next?

After you create a CA object, you can select it from the Policy tree, and then view important information and manage various settings.

- Click the **General** tab to view and modify log and permissions settings.
  - Click the **Log** sub-tab to view any logged events that are triggered by the template object.

    > **IMPORTANT** You must have the *Read* permission to view the Log tab.

    For more information about options found on the Log tab, see Viewing log events.
  - On the **Permissions** sub-tab, you can configure the users or groups to whom you want to grant permissions to the new template object.

    Consider managing object permissions via parent objects so that you can take advantage of inheritance. For more information, see Permission inheritance and flow down.

## MSCA–certificate settings

When you define a CA template object, you provide the information Trust Protection Foundation needs to connect to the CA. However, when you associate a CA template object with a specific certificate object, you can define additional settings that are specific to the certificate. These settings are passed to the CA when Trust Protection Foundation renews the certificate.

The MSCA template lets you enable the specific end date feature, which lets users specify expiration (end) dates for certificates requested from the CA so that they do not expire during known freeze periods.

> **IMPORTANT** The end date you specify is ignored if it exceeds the validity period configured by the Microsoft CA administrator.

**TIP** To use the Allow Users to Specify End Date feature, you *might* need to enable the EDITF_ATTRIBUTEENDDATE flag on the CA, which is typically done using the **certutil** command-line tool:

```
certutil -setreg policy\EditFlags +EDITF_ATTRIBUTEENDDATE
```

You would then restart the **certsvc** service.

For more information, refer to the following Microsoft articles:

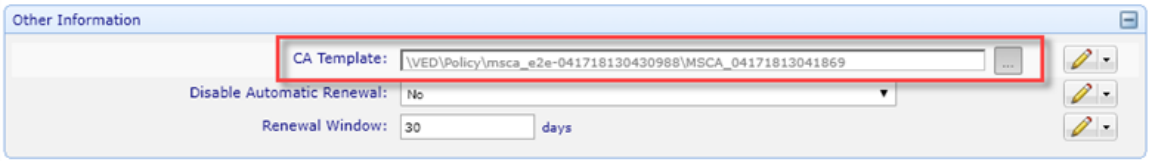
- https://support.microsoft.com/en-us/kb/254632
- https://msdn.microsoft.com/en-us/library/cc226763.aspx

You can verify Microsoft CA settings by entering the following at a command prompt:

```
certutil -getreg policy\EditFlags
```

### To enable and configure specific expiration dates for your MSCA certificates

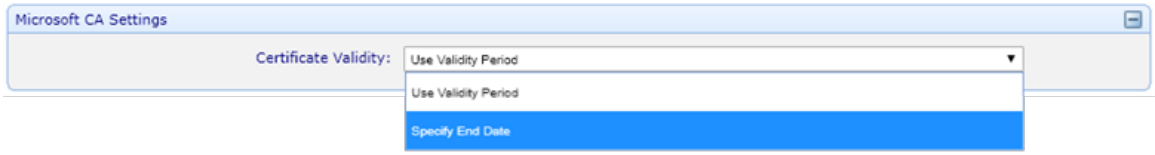
1. From the Platform menu bar, click Policy Tree and either create a new Microsoft CA template, or open an existing Microsoft CA template.
2. In the **Options** box, select the **Allow Users to Specify End Date** check box.
3. When you finish making any other changes to the template, click **Save**.
4. In the Policy tree, either create a new certificate object or open an existing certificate object.
5. In the **Other Information** box, select the CA template you edited in a previous step.



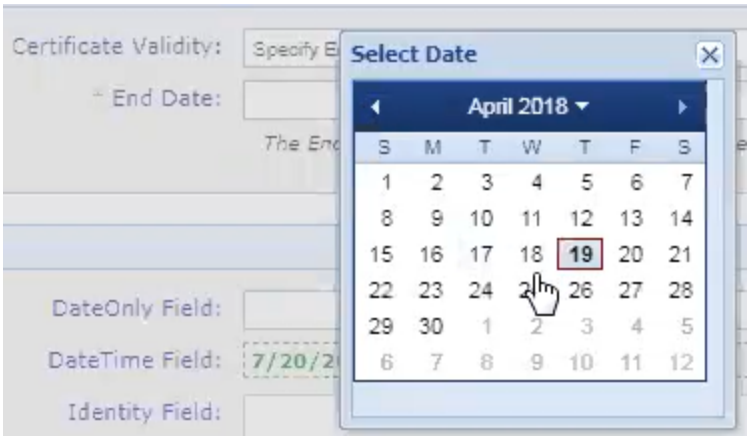

**NOTE** When you select a CA template where you have enabled the end date option, then the **Microsoft CA Settings** box appears.

The box does not appear on the certificate object page unless you have enabled the option in the CA template you select.

6. In the **Microsoft CA Settings** box, select **Specify End Date** from the **Certificate Validity** list.

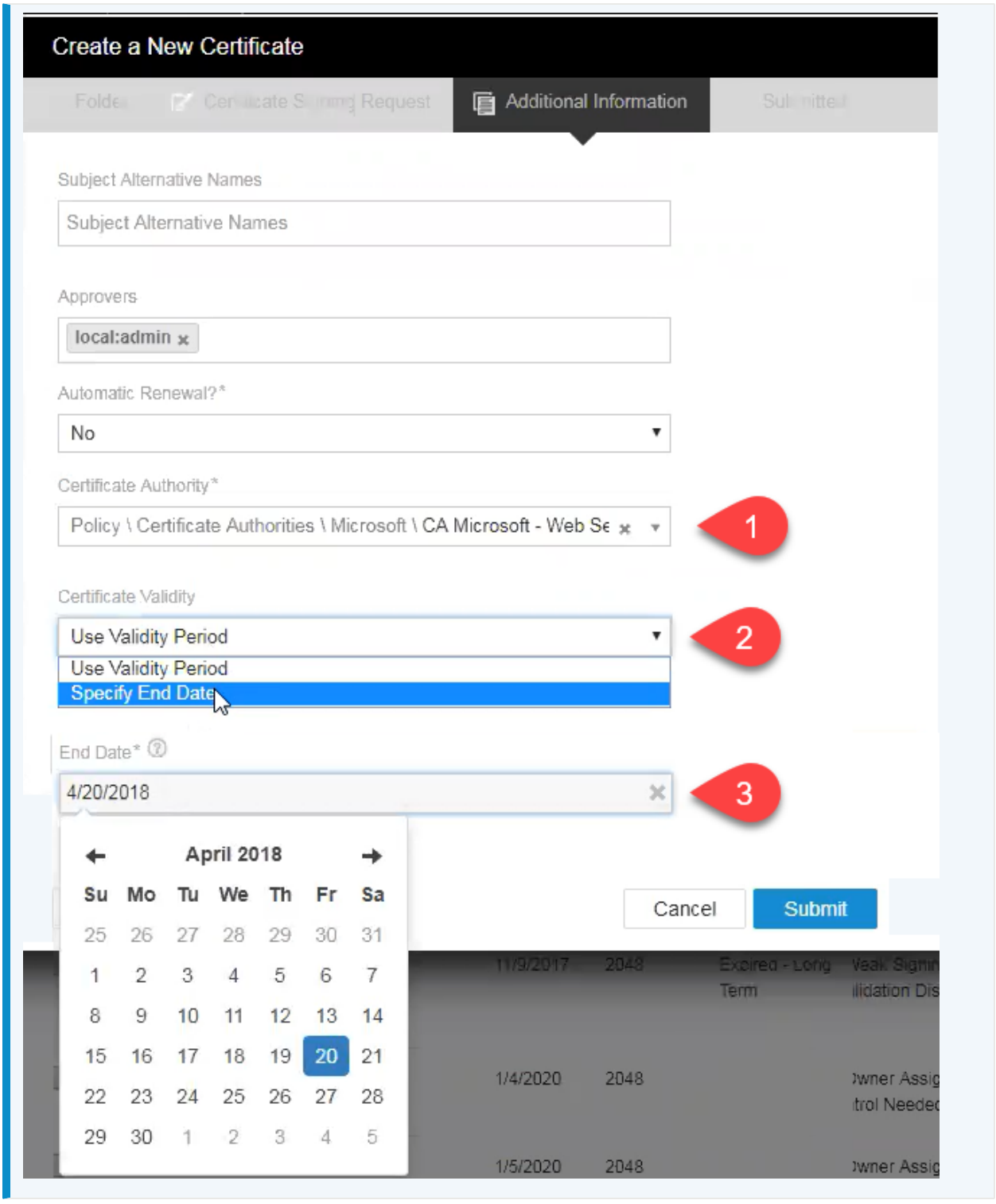



7. Click the **End Date** field and choose an expiration date from the pop-up calendar for certificates requested from the CA so that they don't expire during your known freeze periods.




8. When you finish making any other changes to the certificate object, click **Save**.

**DID YOU KNOW?** You can also select the CA template and configure the end date from main Certificate Manager - Self-Hosted interface, either on the Additional Information page of Renewal Details, or when creating a new certificate:


Create a New Certificate
Additional Information
Subject Alternative Names
Subject Alternative Names
Approvers
local:admin
Automatic Renewal?*
No
Certificate Authority*
Policy \ Certificate Authorities \ Microsoft \ CA Microsoft - Web Se
1
Certificate Validity
Use Validity Period
2
Use Validity Period
Specify End Date
End Date*
4/20/2018
3
April 2018
Su Mo Tu We Th Fr Sa
25 26 27 28 29 30 31
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 1 2 3 4 5
Cancel
Submit
11/9/2017
2048
Expired - Long Term
1/4/2020
2048
1/5/2020
2048

# OpenSSL

This section provides steps required to set up the OpenSSL CA.

This chapter contains the following topics:

# Setting up the OpenSSL certificate authority

OpenSSL provides built-in certificate authority functions that are useful in test environments where security might be less critical. The OpenSSL CA is purely file-based, relying heavily on a configuration file that defines much of its behavior. For details about the OpenSSL CA or to download the software, visit http://www.openssl.org/docs/apps/ca.html.

This topic offers general guidelines for setting up an OpenSSL CA.

## Before you begin

Before you begin to set up a new OpenSSL CA, verify that the OpenSSL software has been properly installed on the Linux server where the CA will be hosted. In addition, verify that the user account used by Trust Protection Foundation to log into the hosting system is authorized and has sufficient permissions to both execute the OpenSSL binary and to create or modify files within the CA home directory.

## Setting up the OpenSSL CA

Setting up a new OpenSSL CA is as simple as creating a directory structure and a few supporting files. You can even host multiple OpenSSL CAs on the same computer system.

### To create multiple OpenSSL CAs on the same computer

```
mkdir /opt/openssl_ca
mkdir /opt/openssl_ca/certs
mkdir /opt/openssl_ca/crl
mkdir /opt/openssl_ca/private
echo 0100 > /opt/openssl_ca/serial
touch /opt/openssl_ca/index
```

The most critical file for the CA is the OpenSSL configuration file, which is generally located in the CA's home directory (for example, /opt/openssl_ca) and is called *openssl.cnf*.

### Sample configuration file taken from a fully functional OpenSSL CA

```
HOME                    = .
RANDFILE                = $ENV::HOME/.rnd

[ ca ]
```

```
default_ca                = CA_default

[ CA_default ]
dir                       = /opt/openssl_ca
crl_dir                   = $dir/crl
database                  = $dir/index
new_certs_dir             = $dir/certs
serial                    = $dir/serial
certificate               = $dir/issuer.crt
private_key               = $dir/private/issuer.key
policy                    = policy_match
default_days              = 365           # 1 year
default_crl_days          = 7             # 7 days
default_md                = sha1
default_bits              = 2048
preserve                  = no
unique_subject            = no
x509_extensions           = v3_req
copy_extensions           = copy          # to enable SubjectAltName

[ policy_match ]
countryName               = optional
stateOrProvinceName       = optional
localityName              = optional
organizationName          = supplied
organizationalUnitName    = optional
commonName                = optional

[ req ]
distinguished_name        = req_distinguished_name

[ req_distinguished_name ]
countryName               = Country (2 letter code)
countryName_min           = 2
countryName_max           = 2
stateOrProvinceName       = State or Province (spelled out)
```

```
localityName                = City or Locality
organizationName            = Organization
organizationalUnitName      = Organizational Unit
commonName                  = Common Name (FQDN)
commonName_max              = 64

[ v3_req ]
basicConstraints            = CA:FALSE
subjectKeyIdentifier        = hash
authorityKeyIdentifier      = keyid,issuer
keyUsage                    = digitalSignature,keyEncipherment
extendedKeyUsage            = serverAuth,clientAuth
crlDistributionPoints       = URI:http://pki.venafi.example/issuer.crl

[ v3_ca ]
basicConstraints            = CA:TRUE
subjectKeyIdentifier        = hash
authorityKeyIdentifier      = keyid:always,issuer:always
keyUsage                    = cRLSign,keyCertSign
```

With the directory structure and supporting files in place, the actual certificate authority can be initialized by creating a key pair and using it to generate a self-signed root CA certificate. You will be prompted to enter a private key password when the key pair is generated and then again when the private key is used to sign the certificate.

If present in the OpenSSL configuration file, the value specified for the string_mask parameter must be one that supports printable strings. These include default, pkix, and nombstr.

### To generate a new key pair

```
openssl genrsa -des3 -out /opt/openssl_ca/private/issuer.key 2048

openssl req -new -x509 -config /opt/openssl_ca/openssl.cnf -extensions v3_ca
-days 4383 \
            -key /opt/openssl_ca/private/issuer.key -out /opt/openssl_
ca/issuer.crt \
            -subj "/CN=FOR TESTING PURPOSES ONLY"
```

## Configuring the OpenSSL template object

To enable Trust Protection Foundation to manage OpenSSL certificates, you must configure the OpenSSL template object. This object provides the information Trust Protection Foundation needs to request, retrieve, and install certificates issued by the OpenSSL driver.

> **BEST PRACTICE** Consider managing CA Template object settings using a policy. For more information, see "Managing CA templates using policies" on page 58 in the *Certificate Authority and Hosting Platform Integration Guide*.

### To create a OpenSSL CA template object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. From the **Tree** drop-down menu, click **Policy**.
3. In the **Policy** tree, select the folder where you want to create the CA Template object, and then click **Add**.
4. Click **CA Template**, then select OpenSSL to create it.
5. In the **CA Name** box, type a name for the new OpenSSL object.
6. Refer to the following table to complete the remaining CA template settings:

#### OpenSSL CA Template Settings

| Field | Description |
|---|---|
| **Connection** | |
| Hostname | IP Address or DNS name of the OpenSSL CA server.<br>Trust Protection Foundation supports both IPv4 or IPv6 connections. |
| Credentials | User name credential required to connect to the OpenSSL CA.<br>The credential type can be:<br>◦ Username |

| Field | Description |
|---|---|
| | ◦ Private Key<br><br>**To select a credential**<br>a. Click the **Browse** button.<br>The **Credential Selector** dialog appears.<br>b. Select the user name credential that stores the user name and password required to connect to the OpenSSL CA server, and then click **Select**. |
| SSH Port | The network port used by the Secure Shell (SSH) protocol for secure remote access to servers. Default port is 22. |
| **Options** | |
| CA Template<br>(Optional) | The OpenSSL CA template that you want to associate with the current CA Template object.<br>**TIP** If you populate the CA template field, it will be used. If left blank, the system will refer to the configuration file and apply the default CA. |
| CA Config File Path | The location of the configuration file for the Certificate Authority (CA) that defines its settings and operations. |
| CA Certificate Path | The location of the public certificate used by the CA to sign other certificates. |

| Field | Description |
|---|---|
| CA Private Key Path | The file path where the private key for the CA is stored, used for signing certificates. |
| Private Key Credential<br>(Optional) | The passphrase or password used to secure the private key. |
| Temp Directory<br>(Optional) | A folder for temporarily storing files during the certificate generation or signing process, |

7. You have two options for your settings, the default CA or the CA Template:
   a. If you choose the default CA, then fill out the required fields (Hostname, Credentials, and SSH Port).
   b. If you choose a CA Template, then fill out the fields (Hostname, Credentials, SSH Port, and enter you CA Template name).
8. Once you have completed the settings fields, Click **Validate**.
9. Click **Save**.

## What's next?

After you create a CA object, you can select it from the Policy tree, and then view important information and manage various settings.

- Click the **General** tab to view and modify log and permissions settings.
  - Click the **Log** sub-tab to view any logged events that are triggered by the template object.

    **IMPORTANT** You must have the *Read* permission to view the Log tab.

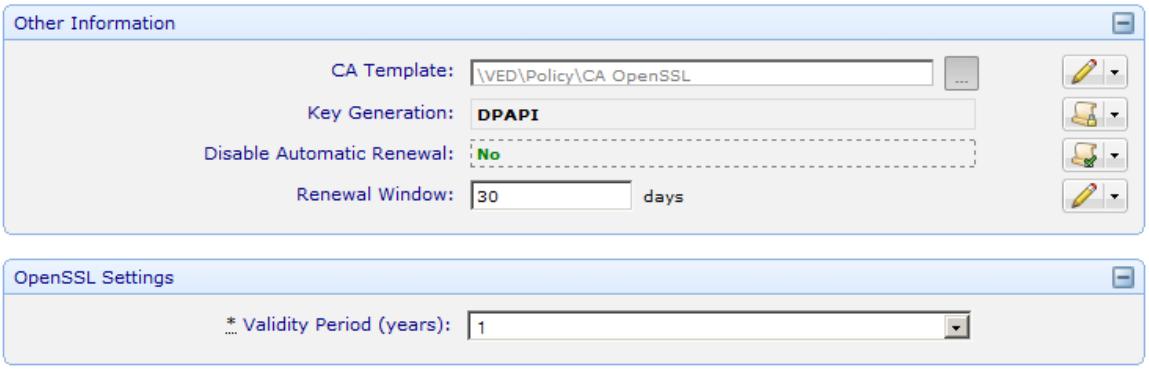
    For more information about options found on the Log tab, see Viewing log events.
  - On the **Permissions** sub-tab, you can configure the users or groups to whom you want to grant permissions to the new template object.

Consider managing object permissions via parent objects so that you can take advantage of inheritance. For more information, see Permission inheritance and flow down.

## OpenSSL settings–certificate object settings

When you define a CA Template object, you provide the information Trust Protection Foundation needs to connect to the CA.

However, when you associate the OpenSSL CA Template object with a specific certificate object, Trust Protection Foundation presents an additional settings field that applies to that specific certificate. Therefore, when you select the CA Template on the certificate object configuration page, you must select the validity period (in years) in the **OpenSSL Settings** area.




The following table reviews the OpenSSL settings in the certificate object configuration.

| Field | Description |
|---|---|
| Validity Period (years) | Indicates the number (in years) in which the certificate will be valid, beginning on the date when the certificate is enrolled.<br><br>The options available in this menu are determined by the Validity Periods configured in the OpenSSL CA template assigned to the certificate object. |

# OpenTrust

The following sections review the steps required to set up the OpenTrust CA.

This chapter contains the following topics:

## Configuring the OpenTrust CA Template object

Trust Protection Foundation can interface with the OpenTrust CA in two ways:

- Using the *Enrollment Entity* method by which Trust Protection Foundation submits requests to, and retrieves certificates from, the OpenTrust CA. However, this method requires that the requests themselves be approved outside of Trust Protection Foundation.
- Using the *Registration Authority* method by which Trust Protection Foundation approves the certificate requests that it has submitted so that the enrollment process is fully automated.

**IMPORTANT** The Windows server where you are running Trust Protection Foundation must trust the certificate chain that secures the OpenTrust web portal and the RA certificate (if applicable). Therefore, it might be necessary to add those root and intermediate certificates to the CAPI trust store.

To enable Trust Protection Foundation to manage OpenTrust certificates, you must configure the OpenTrust CA Template object. This object provides the information that Trust Protection Foundation needs to request, retrieve, and install certificates issued by the OpenTrust CA.

**BEST PRACTICE** Consider managing CA Template object settings using a policy. For more information, see "Managing CA templates using policies" on page 58 in the *Certificate Authority and Hosting Platform Integration Guide*.

### To create and configure a OpenTrust CA template

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. From the **Tree** drop-down menu, click **Policy**.
3. In the **Policy** tree, select the folder where you want to create the CA Template object, and then click **Add**.
4. Click **CA Template**, then select **OpenTrust Enterprise PKI** to create it.
5. In the **CA Name** box, type a name for the new OpenTrust Enterprise PKI object.

Refer to the following table to complete the remaining CA template settings:

| Field | Description |
|---|---|
| **Connection** | |
| Server | There are two choices for this field. Choose either **Enrollment Entity** or **Registration Authority**.<br><br>If Enrollment Entity is selected, a manual approval will need to be done from the OpenTrust CA web portal during provisioning (at step 700). |
| PKI Server | The fully qualified domain name (FQDN) of your OpenTrust CA web portal. |
| Port | The TCP port where your OpenTrust CA web portal is listening for secure connections. The default is 443. |
| Credential | Only required if Registration Authority has been chosen for Server. With Enrollment Entity, assigning this credential could make additional Certificate Profiles available. Select a certificate credential object where you have imported the PKCS#12 certificate provided by OpenTrust. |
| Validate | Click to validate connectivity and authentication. When successful, the Certificate Profiles available from the OpenTrust CA web portal are populated. |
| **Options** | |
| Certificate Profiles | The names of the profiles that are retrieved from the OpenTrust CA web portal. Select one of the Certificate Profiles listed in this field. |
| Subject Alt Name Enabled | Configures the current CA template object to support CSRs with DNS-based subject alt name (SAN) values.<br><br>**NOTE** DNS SANs are included in the Unified Communications Certificates that are used with Microsoft Exchange 2007 and Microsoft Office Communications Server.<br><br>If you do not select this option, the current CA template object will not accept CSRs with SAN values. If Trust Protection Foundation attempts to submit a CSR with SAN values, the CA Template object returns the following error: |

| Field | Description |
|---|---|
| | Stage 400 (Creating CSR) SubjectAltName not supported by Application and CA.<br><br>**IMPORTANT** Before selecting this option, you must also verify the SAN feature is enabled on your CA engine. It it isn't, the CA returns the same error message when Trust Protection Foundation attempts to submit a CSR with SAN values.<br><br>For Secure Email (S/MIME) products, at least one Email SAN is required in the certificate request.<br><br>For additional information on defining SAN values in the certificate, see Subject Alt Name in the topic Certificate settings. |
| **Certificate Fields** | |
| Select | Indicates whether a field without a mapped attribute is visible to users so that they can assign a value for each certificate. |
| Optional | Indicates whether a value must be entered into the field for each certificate being enrolled.<br><br>This field is automatically populated when a certificate profile is selected and is not configurable. |
| Name | The name of the certificate field as it is defined by the OpenTrust CA web portal.<br><br>This field is automatically populated when a Certificate Profile is selected and is not configurable. |
| Display Name | The name of the field as it appears to users who have the CA template assigned to their certificate. |
| Mapped Attribute | The name of the Subject DN or SAN attribute to which the field is mapped. Selectable values include: X509 Subject (CN), X509 SubjectAltName DNS, Organization, Organizational Unit, City, State, Country. |

| Field | Description |
|---|---|
| Default | The initial value of the field that a user sees when the OpenTrust CA template has been assigned to a certificate.<br><br>This field is automatically populated when a Certificate Profile is selected and is not configurable. |
| Editable | Indicates whether a user can override the default value.<br><br>This field is automatically populated when a Certificate Profile is selected and is not configurable. |
| Fixed | Indicates whether the default value can be changed.<br><br>This field is automatically populated when a Certificate Profile is selected and is not configurable. |
| **Validity Period** | |
| Validity Term | The amount of time (in days or years) for which this certificate will be issued. |
| Term Unit | Specifies whether the numeric value entered in the Validity Term is expressed in days or years. |

6. (Optional) To see additional attributes, review the settings on the **Support** tab.
7. Click **Save**.

## What's next?

After you create a CA object, you can select it from the Policy tree, and then view important information and manage various settings.

- Click the **General** tab to view and modify log and permissions settings.
  - Click the **Log** sub-tab to view any logged events that are triggered by the template object.

    **IMPORTANT** You must have the *Read* permission to view the Log tab.

    For more information about options found on the Log tab, see Viewing log events.

- On the **Permissions** sub-tab, you can configure the users or groups to whom you want to grant permissions to the new template object.

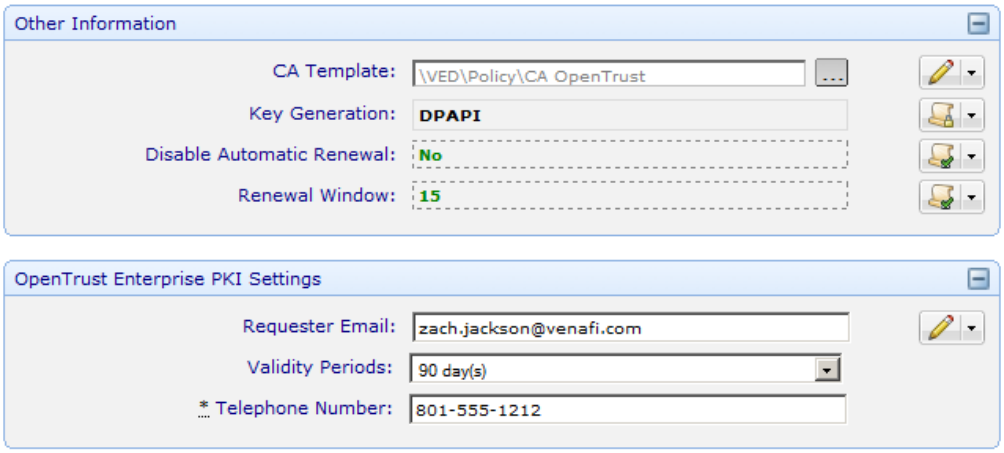
  Consider managing object permissions via parent objects so that you can take advantage of inheritance. For more information, see Permission inheritance and flow down.

## OpenTrust settings–certificate object settings

When you define a CA Template object, you provide the information Trust Protection Foundation needs to connect to the CA. However, when you associate the OpenTrust CA Template object configuration with a specific certificate object, you must define additional settings that are specific to the certificate–information such as the Requester Email, Validity Period, and OpenTrust Certificate Fields. These settings are passed to the OpenTrust CA when Trust Protection Foundation renews the certificate.

When you select the CA Template in the certificate object configuration, you must complete the settings in the OpenTrust Enterprise PKI Settings section.

- In the Policy Tree, the OpenTrust CA settings are presented in the certificate object configuration page when you select the OpenTrust CA Template object.




The following table reviews the OpenTrust settings in the certificate object configuration.

| Field | Policy | Description |
|---|---|---|
| **OpenTrust Enterprise PKI Settings** | | |
| Requester Email | Yes | Email address of the person making the certificate enrollment request. |

| Field | Policy | Description |
|---|---|---|
| | | |
| Validity Periods | No | Period of time (in days or years) that the certificate is valid.<br><br>The options available in this menu are determined by the Validity Periods configured in the OpenTrust CA Template object configuration. For more information, see "Configuring the DigiCert CA template object" on page 115 |
| OpenTrust Certificate Fields | No | Zero or more special fields that are defined for the Certificate Profile selected on the OpenTrust CA Template object that is assigned to the certificate object. |

# QuoVadis

In order to use the QuoVadis CA driver, your account needs to be prepared by QuoVadis for automated access.

### To prepare your account for automated access

1. Obtain a Web Service Signing Certificate (Web API Certificate) issued by QuoVadis.

   This certificate is used by the driver to access the QuoVadis API.

2. Provide QuoVadis with the IP addresses of those Trust Protection Foundation servers that will be connecting to QuoVadis's APIs.

   For security reasons QuoVadis blocks API access from any address that is not registered. If NAT services are used on your network, you must register your external address.

3. Associate your Web Service Signing Certificate in QuoVadis with your IP address, and then configure firewall and web service access.

4. Using the QuoVadis web service, select which certificate templates can be requested and configure QuoVadis to enable these, accordingly.

   To protect your organization, only templates that you specifically request to be enabled can be used via the web service.

5. Make sure that a subscriber is registered to the account and assigned to the organizations that are going to request certificates through the web service.

The following topics include the steps required to set up the QuoVadis CA:

This chapter contains the following topics:

## Configuring the QuoVadis CA template object

To enable Trust Protection Foundation to manage QuoVadis certificates, you must configure the QuoVadis CA Template object. This object provides the information that Trust Protection Foundation needs to request, retrieve, renew and revoke certificates issued by the QuoVadis CA.

**BEST PRACTICE** Consider managing CA Template object settings using a policy. For more information, see "Managing CA templates using policies" on page 58 in the *Certificate Authority and Hosting Platform Integration Guide*.

### To create and configure a QuoVadis CA template

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. From the **Tree** drop-down menu, click **Policy**.
3. In the **Policy** tree, select the folder where you want to create the CA Template object, and then click **Add**.
4. Click **CA Template**, then select **QuoVadis CA** to create it.
5. In the **CA Name** box, type a name for the new QuoVadis CA object.
6. Refer to the following table to complete the remaining CA template settings:

#### QuoVadisCA Template Configuration

| Field | Description |
|---|---|
| **Connection** | |
| Web API Certificate | The certificate credential issued by the QuoVadis CA, which authenticates access to interact with the QuoVadis web service API. |
| Subscriber Email | The RFC822 formatted email address of the subscriber account registered with QuoVadis and that will be used to request certificates. |
| Account Name | The name of the customer account that is established and provided by QuoVadis. |

| Field | Description |
|---|---|
| Account Organization | The name of the customer organization (or department) that has been approved for enrollment by QuoVadis. |
| Subject Alt Name Enabled | Configures the current CA template object to support CSRs with DNS-based subject alt name (SAN) values.<br><br>**NOTE** DNS SANs are included in the Unified Communications Certificates that are used with Microsoft Exchange 2007 and Microsoft Office Communications Server.<br><br>If you do not select this option, the current CA template object will not accept CSRs with SAN values. If Trust Protection Foundation attempts to submit a CSR with SAN values, the CA Template object returns the following error:<br><br>Stage 400 (Creating CSR) SubjectAltName not supported by Application and CA.<br><br>**IMPORTANT** Before selecting this option, you must also verify the SAN feature is enabled on your CA engine. It it isn't, the CA returns the same error message when Trust Protection Foundation attempts to submit a CSR with SAN values.<br><br>For Secure Email (S/MIME) products, at least one Email SAN is required in the certificate request.<br><br>For additional information on defining SAN values in the certificate, see Subject Alt Name in the topic Certificate settings. |
| Use test account | |

| Field | Description |
|---|---|
| Validate | Click this button to validate the connection to the QuoVadis CA and to retrieve the Policy Templates that are available for enrollment. |
| **Options** | |
| Template | The Policy Template for which certificates will be requested during enrollment. Policy Templates define Validity Period, Subject DN behavior, support for Subject Alternative Names (SANs), and the certificate type (e.g. Business SSL, EV SSL, etc.). |

7. (Optional) To see additional attributes, review the settings on the **Support** tab.
8. Click **Save**.

### What's next?

After you create a CA object, you can select it from the Policy tree, and then view important information and manage various settings.

- Click the **General** tab to view and modify log and permissions settings.
  - Click the **Log** sub-tab to view any logged events that are triggered by the template object.

    **IMPORTANT** You must have the *Read* permission to view the Log tab.

    For more information about options found on the Log tab, see Viewing log events.
  - On the **Permissions** sub-tab, you can configure the users or groups to whom you want to grant permissions to the new template object.

    Consider managing object permissions via parent objects so that you can take advantage of inheritance. For more information, see Permission inheritance and flow down.

## QuoVadis Settings–Certificate Object Settings

When you define a CA Template object, you provide the information that Trust Protection Foundation needs to connect to the CA as well as several other properties of the certificate to be enrolled.

However, the QuoVadis CA template currently does not offer any settings that are specific to the certificate object to which it has been applied.

## RedHat Certificate System

The following section reviews the steps required to set up the RedHat CA Template object so CyberArk Trust Protection Foundation™ can initiate and auto-enroll new or to-be-renewed certificate and key generation requests with the RedHat CA:

This chapter contains the following topics:

## Configuring the RedHat Certificate System CA template object

To enable Trust Protection Foundation to manage RedHat Certificate System certificates, you must configure the RedHat Certificate System CA Template object. This object provides the information Trust Protection Foundation needs to request, retrieve, and install certificates issued by the RedHat Certificate System CA.

> **BEST PRACTICE** Consider managing CA Template object settings using a policy. For more information, see "Managing CA templates using policies" on page 58 in the *Certificate Authority and Hosting Platform Integration Guide*.

### To create and configure a RedHat Certificate System CA template

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. From the **Tree** drop-down menu, click **Policy**.
3. In the **Policy** tree, select the folder where you want to create the CA Template object, and then click **Add**.
4. Click **CA Template**, then select **RedHat** to create it.
5. In the **CA Name** box, type a name for the new RedHat object.

Refer to the following table to complete the remaining CA template settings:

### RedHat Certificate System CA Template Configuration

| Field | Policy | Description |
|---|---|---|
| **Connection** | | |
| Credential | No | Certificate Credential for the administrator certificate and private key that is required to connect to the RedHat Certificate System CA server. |
| Hostname | No | IP address or hostname of the server where RedHat Certificate SystemCertificate Manager - Self-Hosted is installed.<br><br>Trust Protection Foundation supports both IPv4 or IPv6 connections. |
| End Entity URL Suffix | No | Suffix identifying the subsystem that hosts the RedHat Certificate System End Entity interface.<br><br>▪ **ca** is for the certificate authority |

| Field | Policy | Description |
|---|---|---|
| | | ■ **ra** is for the registration authority |
| End Entity Port | No | Port where the End Entity service listens for PKI requests, such as enrollment, renewal, and revocation of certificates. |
| Agent URL Suffix | No | Suffix identifying the subsystem that hosts the RedHat Certificate System Agent interface.<br>■ **ca** is for the certificate authority<br>■ **ra** is for the registration authority |
| Agent Port | No | Port where the Agent service listens for certificate issuance requests. |
| **Options** | | |
| Enrollment Mode | No | Profile or policy you want to associate with the current CA Template object. Trust Protection Foundation references this profile or policy when it submits the CSR. |
| Use Profile | No | Allows the administrator to associate a profile with the current RedHat Certificate System CA Template object.<br>A certificate profile defines everything associated with the issuance of a particular type of certificate. The certificate that is issued from a certificate profile request contains the default content with values derived from the parameters associated with those defaults.<br>For example, you could set up a certificate profile for an SSL certificate that defines all aspects of that certificate including a default validity period of two years. When a user sends a request associated with this certificate profile, the issued certificate contains the information specified in the profile defaults and is valid for two years.<br>**To add a profile to the RedHat Certificate System CA profile**<br>Select **Use Profile**.<br>Click **Load**.<br>Trust Protection Foundation automatically downloads the profiles from the RedHat Certificate System CA. |

| Field | Policy | Description |
|---|---|---|
| | | In the Profile drop-down list, select the profile you want to associate with the current RedHat Certificate System CA Template object. |
| Use Policy | No | Allows the administrator to associate a policy with the current RedHat Certificate System CA Template object.<br><br>A certificate policy is a set of rules that Certificate Services uses to evaluate or verify a certificate request. This includes rules for evaluating certificate formulation, signing, and renewal requests. For example, you can configure a Certificate Manager - Self-Hosted's policy to impose restrictions on validity length, key type, key length, subject name, extensions, and signing algorithm during certificate issuance.<br><br>To associate the current RedHat Certificate System CA Template object with a policy, simply type an existing policy name in the **Policy** field. |

| Field | Policy | Description |
|---|---|---|
| Manual Approvals | No | Requires manual approval for all CSRs submitted using the current CA Template object.<br><br>When you enable **Manual Approval**, the administrator must log in to the MicrosoftDigiCert CertCentralEntrust Certificate Services CA service and manually approve the renewing certificate.<br><br>If both **Manual Approval** and **Subject Alt Names** are enabled on the CA Template object, Trust Protection Foundation does not include the SAN values in the CSR. Instead, it notifies the approver to provide the DNS SAN values when approving the certificate. |
| Subject Alt Name Enabled | No | Configures the current CA template object to support CSRs with DNS-based subject alt name (SAN) values.<br><br>**NOTE** DNS SANs are included in the Unified Communications Certificates that are used with Microsoft Exchange 2007 and Microsoft Office Communications Server.<br><br>If you do not select this option, the current CA template object will not accept CSRs with SAN values. If Trust Protection Foundation attempts to submit a CSR with SAN values, the CA Template object returns the following error:<br><br>`Stage 400 (Creating CSR) SubjectAltName not supported by Application and CA.`<br><br>**IMPORTANT** Before selecting this option, you must also verify the SAN feature is enabled on your CA engine. It it isn't, the CA returns the same error message when Trust Protection Foundation attempts to submit a CSR with SAN values.<br><br>For Secure Email (S/MIME) products, at least one Email SAN is required in the certificate request.<br><br>For additional information on defining SAN values in the certificate, see Subject Alt Name in the topic Certificate settings. |

6. (Optional) To see additional attributes, review the settings on the **Support** tab.
7. Click **Save**.

## What's next?

After you create a CA object, you can select it from the Policy tree, and then view important information and manage various settings.

- Click the **General** tab to view and modify log and permissions settings.
  - Click the **Log** sub-tab to view any logged events that are triggered by the template object.

    **IMPORTANT** You must have the *Read* permission to view the Log tab.

    For more information about options found on the Log tab, see Viewing log events.
  - On the **Permissions** sub-tab, you can configure the users or groups to whom you want to grant permissions to the new template object.

    Consider managing object permissions via parent objects so that you can take advantage of inheritance. For more information, see Permission inheritance and flow down.

# Sectigo Certificate Manager (SCM)

In order for CyberArk Trust Protection Foundation™ to interact with the Sectigo Certificate Manager (SCM), SCM it is recommended to use a Sectigo API administrator account with the following settings configured in Sectigo Certificate Manager:

- Sectigo admin type: API
  Refer to Adding administrators.
- Create a Sectigo API key
  Refer to Managing admin API keys.
- Certificate requests: Enable **Automatically approve certificate requests** under **Edit Standard Admin**.
  Refer to Managing administrators.

**NOTE** If using Trust Protection Foundation versions 24.3 or lower, use a Sectigo Standard administrator account in Sectigo Certificate Manager.

**NOTE** Some product types, such as EV, might require approval using the SCM web portal.

## Configuring the Sectigo Certificate Manager CA template object

To enable Trust Protection Foundation to manage certificates from the Sectigo Certificate Manager (SCM) web service, you must configure the SCM CA template object. This object provides the information Trust Protection Foundation needs to request, retrieve, and install certificates issued by the Sectigo CA.

**NOTE** In Trust Protection Foundation 26.1 and later, Entrust Certificate Services is no longer supported. Revocation of certificates originally issued by Entrust is handled by the Sectigo Certificate Manager driver.

In addition to managing Sectigo-issued certificates, a Sectigo CA template is required to revoke legacy certificates originally issued by Entrust.

**BEST PRACTICE** Consider managing CA Template object settings using a policy. For more information, see "Managing CA templates using policies" on page 58 in the *Certificate Authority and Hosting Platform Integration Guide*.

### To create and configure a Sectigo Certificate Manager CA template

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. From the **Tree** drop-down menu, click **Policy**.
3. In the **Policy** tree, select the folder where you want to create the CA Template object, and then click **Add**.
4. Click **CA Template**, then select **Sectigo** to create it.
5. In the **CA Name** box, type a name for the new Sectigo object.

6. Refer to the following table to complete the remaining CA template settings:

| Field | Description |
|---|---|
| **Connection** | |
| Credentials | Username Credential required to connect to the Sectigo Certificate Manager web service.<br><br>**To select the Credential**<br>a. Click the **Browse** button.<br>b. In the **Credential Selector** dialog box, select the Username Credential that stores the username and password required to connect to the SCM web service, and then click **Select**. |
| Sectigo Certificate Manager URL | URL that you use to access the Sectigo Certificate Manager web portal. This value is used by Trust Protection Foundation to determine which SCM server hosts your account and to identify the customer name you were assigned by Sectigo. For example, if your SCM URL is https://cert-manager.com/customer/mycompany, then your account is hosted by the cert-manager.com server and your customer name is "mycompany". Both variables are necessary to successfully interface with the SCM API. |
| Organization ID | The numeric code that Sectigo has assigned to your organization. This read-only value is found in the Sectigo Certificate Manager web portal by navigating to the Settings > Organizations section, selecting the organization, clicking the Edit button, and then selecting the SSL tab.<br><br>You must also ensure the Web API option is checked in the Sectigo Certificate Manager web portal for this organization. |
| Organization Secret Key | The string of characters that has been assigned to your organization for restricting enrollment. This editable value is found in the Sectigo Certificate Manager web portal by navigating to the Settings > Organizations section, selecting the organization, clicking the Edit button, and then selecting the SSL tab. |
| Validate | Tests the selected Username Credential to ensure Trust Protection Foundation can authenticate with the SCM web service. |

| Field | Description |
|---|---|
| | If the credential is valid and the connection succeeds, Trust Protection Foundation automatically populates the available Product Types. |
| **Options** | |
| Product Type | Sectigo certificate product that you want to associate with the current CA Template object. Trust Protection Foundation references this template when it submits the CSR. Click the **Product Type** drop-down list to select the template you want to associate with the current CA Template object.<br><br>When you click **Validate** to validate the CA credentials and the connection succeeds, Trust Protection Foundation retrieves the supported templates from the SCM web service.<br><br>Trust Protection Foundation supports at least the following Sectigo certificate profiles:<br><br>◦ Sectigo Unified Communications Certificate<br>◦ Sectigo EV SSL Certificate<br>◦ Elite SSL Certificate<br>◦ Multi-Domain Instant SSL Certificate<br>◦ Premium SSL<br>◦ Premium SSL Wildcard Certificate<br><br>**NOTE** These certificate profiles are not default profile names as profile names and settings are customizable.<br><br>**DID YOU KNOW?** EV certificates are a special type of X.509 certificate that can be issued only by a public CA and which require more extensive investigation by the CA before being issued. |
| SAN Enabled | Configures the current CA template object to support CSRs with DNS-based Subject Alt Name (SAN) values. |

| Field | Description |
|---|---|
| | DNS SANs are included in the Unified Communications Certificates that are used with Microsoft Exchange 2007 and Microsoft Office Communications Server.<br><br>If you do not select this option, the current CA Template object will not accept CSRs with SAN values. If Trust Protection Foundation attempts to submit a CSR with SAN values, the CA Template object returns the following error:<br><br>Stage 400 (Creating CSR) SubjectAltName not supported by Application and CA.<br><br>Before selecting this option, you must also verify the SAN feature is supported by the chosen product type; otherwise, the CA returns the same error message when Trust Protection Foundation attempts to submit a CSR with SAN values.<br><br>For information on defining SAN values in the certificate, see Certificate Object Settings. |
| **Validity Period** | This feature allows you to control which Validity Periods can be selected when configuring a Sectigo certificate.<br><br>The Validity Period is the period of time (in days) that the certificate is valid, with a maximum of 398 days for public certificates. |
| Available Validity Periods (Days) | Validity Periods that are supported by the Sectigo CA for the chosen product type. |
| Selected Validity Periods (Days) | Validity Periods that are available when configuring a Sectigo certificate.<br><br>To populate this list, select the Selected Validity Periods you want to be available when configuring Sectigo certificates in Policy Tree, and then click the right-arrow.<br><br>Press **Shift+click** to select multiple, contiguous items.<br><br>Press **Ctrl+click** to select multiple, discontiguous items. |

7. (Optional) To see additional attributes, review the settings on the **Support** tab.
8. Click **Save**.

## What's next?

After you create a CA object, you can select it from the Policy tree, and then view important information and manage various settings.

- Click the **General** tab to view and modify log and permissions settings.
  - Click the **Log** sub-tab to view any logged events that are triggered by the template object.

    **IMPORTANT** You must have the *Read* permission to view the Log tab.

    For more information about options found on the Log tab, see Viewing log events.
  - On the **Permissions** sub-tab, you can configure the users or groups to whom you want to grant permissions to the new template object.

    Consider managing object permissions via parent objects so that you can take advantage of inheritance. For more information, see Permission inheritance and flow down.

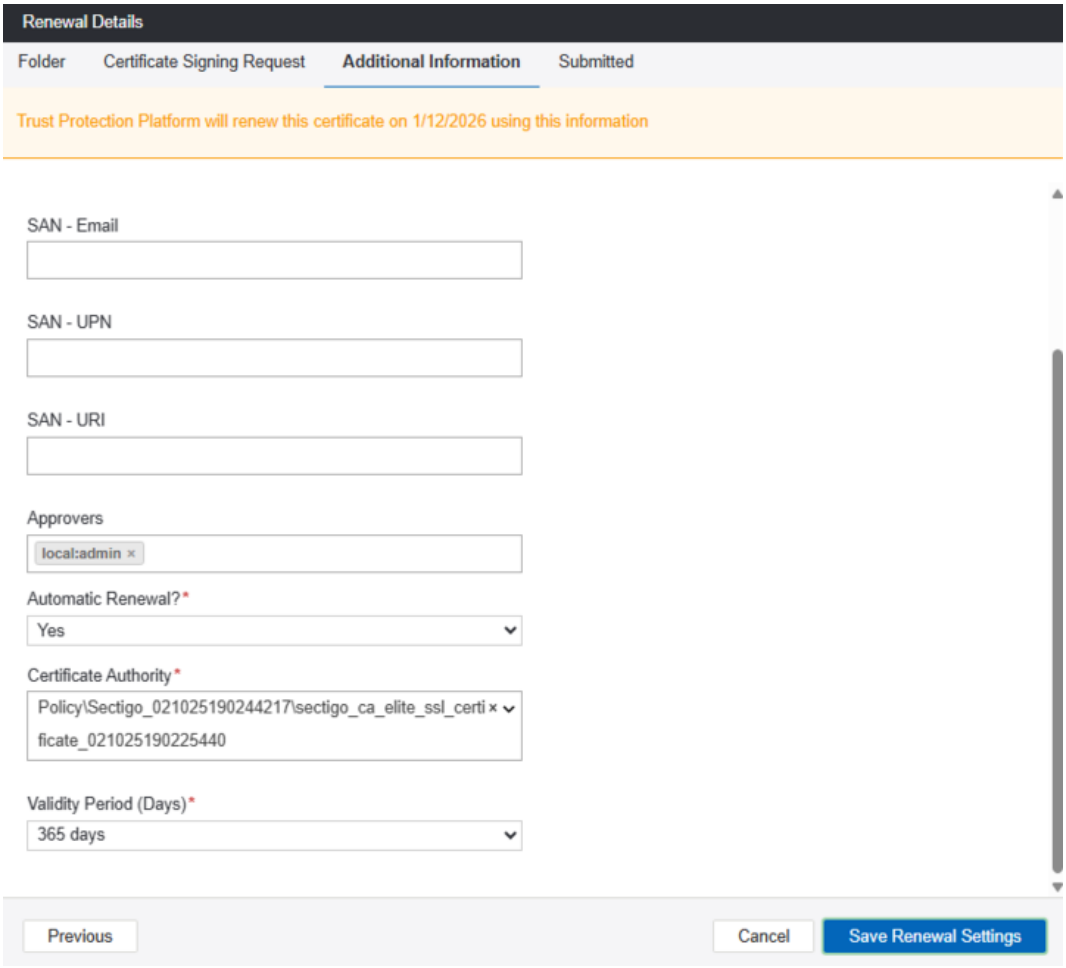
## Sectigo Certificate Manager–certificate settings

When you define a CA template object, you provide the information Trust Protection Foundation needs to connect to the CA.

However, when you associate a Sectigo CA template object with a specific Certificate object, you must define additional settings that are specific to the certificate. These settings are passed to the Sectigo CA when Trust Protection Foundation enrolls the certificate.

**NOTE** Only CyberArk's Symantec Managed PKI for SSL and Entrust Certificate Services drivers support vendor custom fields. If you want to use custom fields, we recommend using CyberArk's custom fields feature. To ensure a smooth ride, either ensure that all custom fields for your chosen vendor are *optional* and use CyberArk's custom fields instead, or use one of CyberArk's custom field-supported drivers (Symantec or Entrust).

The Sectigo settings appear on the Additional Information section of the Renewal Details.




The following table outlines the Sectigo settings in the certificate object configuration.

### Sectigo Settings in the Certificate Object Configuration

| Field | Policy | Description |
|---|---|---|
| **Settings** | | |
| Validity Period (Days) | No | Period of time (in days) that the certificate is valid.<br><br>The options available in this menu are determined by the Available Validity Periods configured in the Sectigo CA template object configuration. For more information, see "Configuring the Sectigo Certificate Manager CA template object" on page 188. |

| Field | Policy | Description |
|---|---|---|
| | | **NOTE** The Validity Period should match one of the Validity Periods supported by the Certificate Profile Template. For additional terms on the CA template, users must contact Sectigo Customer Support to specify the additional "term" values they wish to add and to which CA Product/Template. Sectigo will then add those values on their side. |

## Symantec Managed PKI for SSL

The topics in this section review the steps required to set up the Symantec Managed PKI for SSL (Symantec MPKI) CA.

**IMPORTANT** This driver is for enrolling publicly trusted certificates from Symantec and is not the same as the DigiCert PKI Platform, which is a private PKI solution.

**DID YOU KNOW?** CyberArk supports integration with the following Symantec CA offerings:

- Symantec Local Hosting Kit (Symantec LHK): Used for managing legacy, non-publicly trusted certificates.
- Symantec Managed PKI for SSL: Used for managing publicly trusted certificates. For more information, visit https://www.symantec.com/ssl-certificates/managed-pki-ssl/.
- DigiCert PKI Platform: Used for managing non-publicly trusted certificates. For more information, see "Using the sample DigiCert PKI Platform PowerShell script" on page 65 and visit https://www.symantec.com/products/information-protection/managed-pki-service.

**IMPORTANT** Before you begin, make sure you've reviewed and completed all required prerequisite steps for setting up your CA integration. See "About CA driver prerequisites" on page 17.

This chapter contains the following topics:

## Configuring the Symantec Managed PKI for SSL CA template

To enable Trust Protection Foundation to manage Symantec certificates, you must configure a Symantec Managed PKI for SSL (Symantec MPKI) CA template. This template provides the information Trust Protection Foundation needs to request, retrieve, renew, and revoke certificates issued by the Symantec CA.

Trust Protection Foundation supports all certificate types that are supported by the Symantec VICE2 API.

**BEST PRACTICE** Consider managing CA Template object settings using a policy. For more information, see "Managing CA templates using policies" on page 58 in the *Certificate Authority and Hosting Platform Integration Guide*.

### To create and configure a Symantec MPKI CA template

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. From the **Tree** drop-down menu, click **Policy**.
3. In the **Policy** tree, select the folder where you want to create the CA Template object, and then click **Add**.
4. Click **CA Template**, then select **Symantec MPKI** to create it.
5. In the **CA Name** box, type a name for the new Symantec MPKI object.
6. Refer to the following table to complete the remaining CA template settings:

| Field | Description |
|---|---|
| **Connection** | |
| Credential | Credential required to connect to the Enrollment MPKI URL that Symantec assigns to your organization. |

| Field | Description |
|---|---|
| | Symantec issues an administrator certificate and private key when you submit your enrollment. The Symantec MPKI Administrator certificate credential must have the Web Services Application role, which must by granted by Symantec. Make sure that you specify that the private key is exportable when enrolling for this certificate.<br><br>To select the credential, you must first create a Certificate Credential object and import the .pfx certificate file from the workstation where you enrolled for the Symantec MPKI Administrator Certificate. For more information, see Working with system credentials in the *CyberArk Trust Protection Foundation Administration Guide*.<br><br>**To select the Credential**<br><br>1. Click the **Browse** button.<br><br>2. Select the certificate credential you created with the Symantec MPKI Administrator Certificate, and then click **Select**.<br><br>**NOTE** To enable Trust Protection Foundation to manage Symantec Managed PKI for SSL certificates and templates, the VICE2 administrator certificate only requires the Web Services role. |
| | **To verify that the Symantec MPKI administrator certificate has the proper privileges**<br><br>1. Log in to the Symantec Control Center.<br><br>2. Click **Configuration** > **Administrators**.<br><br>3. Select the administrator, and then click **View/Edit**.<br><br>4. Verify that the required permissions are selected. |
| Organization | Name that uniquely identifies your Organization. This name is included in all certificates issued using the current CA template object. |

| Field | Description |
|---|---|
| | When you validate the credential, the supported validity periods, credit accounting values, and domains are automatically populated.<br><br>**TIP** If you have more than one registered organization, you can select any of the organizations that appear on the Organization drop-down list because you can use the same Trust Protection Foundation CA template object for all of them. However, the list of domains on the CA template page will only show those domains that are registered for the organization you select. Therefore, if you are working across multiple domains that are registered across multiple organizations, then you will likely need to create a CA template for each organization. |
| Connect to Test Service | Configures the CA template to use Symantec Managed PKI for SSL's non-production (pilot) web service for testing purposes.<br><br>When this option is enabled, certificates enrolled using the CA template will not be generally trusted by clients because they are issued by an untrusted certificate authority. |
| **Options** | |
| Template | Symantec MPKI unit type, or template, you want to associate with the current CA template object.<br><br>Trust Protection Foundation references the unit type when it submits the CSR. The unit type determines what type of certificate is requested by the current CA template object.<br><br>Trust Protection Foundation supports the following unit types:<br><br>■ Standard EV SSL<br>■ Premium EV SSL<br>■ Standard SSL<br>■ Premium SSL<br>■ Standard Intranet SSL |

| Field | Description |
| --- | --- |
| | ■ Premium Intranet SSL<br>■ Private SSL<br>■ RapidSSL Enterprise<br>■ OFX SSL<br><br>The list of available unit types is automatically retrieved when you validate the CA credential. Only the SSL unit types available to your Symantec Managed PKI for SSL account are loaded into this list. |
| Allow Users to Specify End Date | Lets users specify expiration (end) dates for certificates requested from the CA so that they do not expire during your known freeze periods.<br><br>Typically, renewing certificates that expire during freeze periods requires a more challenging approval process. Setting expiration (or end) dates that fall outside of freeze periods avoids potential interruptions.<br><br>When a specific end date has been specified, the issued certificate has that date as its expiration date.<br><br>This check box is cleared after successful enrollment (so that the validity period takes effect thereafter).<br><br>In order to support enrollment of certificates with specific end dates, the **Applicants can request a specific end date within the validity period** option must be selected on the **Configuration** > **Enrollment** > **Select Certificate Lifecycle Options** page of the Managed PKI for SSL Control Center. |
| Signature Algorithm | Specifies the algorithm that will be used by Symantec Managed PKI for SSL when it signs a requested certificate. Supported values are SHA1-RSA (default) and SHA256-RSA.<br><br>In order to support the SHA256-RSA signature algorithm, the "Enable Selection of Signature Algorithm" option must be selected on the **Configuration** > **Enrollment** page of the Managed PKI for SSL Control Center. |

| Field | Description |
| --- | --- |
| Subject Alt Name Enabled | Configures the current CA template object to support CSRs with DNS-based subject alt name (SAN) values.<br><br>**NOTE** DNS SANs are included in the Unified Communications Certificates that are used with Microsoft Exchange 2007 and Microsoft Office Communications Server.<br><br>If you do not select this option, the current CA template object will not accept CSRs with SAN values. If Trust Protection Foundation attempts to submit a CSR with SAN values, the CA Template object returns the following error:<br><br>Stage 400 (Creating CSR) SubjectAltName not supported by Application and CA.<br><br>**IMPORTANT** Before selecting this option, you must also verify the SAN feature is enabled on your CA engine. It it isn't, the CA returns the same error message when Trust Protection Foundation attempts to submit a CSR with SAN values.<br><br>For Secure Email (S/MIME) products, at least one Email SAN is required in the certificate request.<br><br>For additional information on defining SAN values in the certificate, see Subject Alt Name in the topic Certificate settings. |
| Validity Periods (Years) | Validity Periods that are available to be selected when configuring a Symantec MPKI certificate.<br><br>Lets you to control which Validity Periods can be selected when configuring a Symantec MPKI certificate. The Validity Period is the period of time (in years) that the certificate is valid.<br><br>To populate this list, select the supported validity periods you want to be available when configuring Symantec MPKI certificates in Policy Tree, and then click the right-arrow.<br><br>Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups. |

| Field | Description |
|---|---|
| Renewal Window (Days) | A renewal window is the time period before a certificate's expiration date during which a certificate can be renewed.<br><br>This option lets you specify the maximum number of days prior to the certificate's expiration date that Trust Protection Foundation should treat as a renewal request. If the number of days remaining in the validity period of the certificate is greater than your renewal window setting, then certificate renewal requests are processed as *reissuance* requests.<br><br>When there is no existing certificate, a new certificate request is submitted to the CA as a new (first time) request.<br><br>**TIP** You can set the renewal window to a maximum of 364 days; the default setting is 90 days. But the renewal window does not *prevent* you from renewing a certificate that is not yet within the renewal window. Instead it allows you to obtain a replacement version of a certificate when it is not close to expiring while keeping the same expiration date. |
| Attempt Renewal | Clear the **Attempt Renewal** checkbox if you want Trust Protection Foundation to always request a *new* certificate whenever a certificate should be renewed. By default, Trust Protection Foundation attempts to renew the certificate. In cases where the renewal fails, Trust Protection Foundation then attempts to enroll a new certificate for the expiring certificate.<br><br>Certificate renewal will fail in cases where the certificate being renewed was not issued by the CA, or in cases where the challenge phrase is different from the one used to enroll the original certificate.<br><br>The benefit of renewal is that the expiration date (month and day) remains the same. However, with enrollment, the expiration date is the date on which the request for a new certificate was made. |
| Allow Reissuance | Check the **Allow Reissuance** box and type a number (in days) in the **Renewal Window** field if you want to allow some certificate renewals to be handled as *reissuance* requests. This setting is disabled (unchecked) by default.<br><br>Allow Reissuance is useful in situations where you need to renew a certificate but do not want to change its original expiration date. |

| Field | Description |
| --- | --- |
| | If you disable Allow Reissuance and a certificate's remaining validity is outside of the Renewal Window, then an error should result indicating the certificate cannot be reissued because the CA template doesn't allow it. If that occurs, then you'll need to request that a Trust Protection Foundation administrator enable Allow Reissuance on the CA template, revoke and then renew the certificate, or create a new certificate object and enroll it using the same CA template (since it would be treated as a first-time enrollment).<br><br>**TIP** If you issue a certificate using the Specific End Date option that is set outside of the renewal window, the certificate is reissued with the expiration date you specified in Specific End Date.<br><br>**EXAMPLE** Suppose you're updating SHA1 certificates to SHA2. However, you know that updating them could result in changing their expiration dates to fall inside of a *freeze period*– a time in which your IT department is verifying network and system stability. To avoid this collision, you enable Allow Reissuance to ensure that the certificate's expiration date remains the same. You accept the default 90-day renewal window because it is longer than your freeze period. |
| Certificate Transparency | Configure how Symantec MPKI should handle Certificate Transparency (CT) logging by choosing one of the following options:<br><br>■ Log complete domain names: choose this (default) option to have your entire fully qualified domain names logged to certificate transparency log servers.<br><br>■ Log with sub-domains redacted: choose this option if the certificate you're enrolling is used internally only. When enabled, the Google Chrome web browser will return a warning that the certificate is untrusted.<br><br>**Why can't I opt out of CT logging?** In order for CT logging to add value, Symantec must be able to log all (100%) certificates. For more information, see http://www.symantec.com/connect/blogs/privacy-redaction-and-certificate-transparency. |

| Field | Description |
|---|---|
| **Accounting** | The Accounting fields enumerate the total number of purchased credits, the number of credits used, and the number of credits remaining. These fields are updated when you click the **Validate** button and when Trust Protection Foundation approves a certificate. |
| Credits Ordered | Number of purchased certificate credits. |
| Credits Used | Number of purchased certificate credits that have been used. |
| Credits Remaining | Number of purchased certificate credits that are available. |
| Credits Expire | Date when the purchased certificate credits expire. |
| Domains | All the Internet domains the current CA Template object can submit CSRs for. |
| Credit Alert | Threshold at which Trust Protection Foundation begins sending certificate credit alert notifications.<br><br>When the number of remaining credits reaches this threshold, Trust Protection Foundation generates credit alert events. |

7. (Optional) To see additional attributes, review the settings on the **Support** tab.
8. Click **Save**.

## What's next?

After you create a CA object, you can select it from the Policy tree, and then view important information and manage various settings.

- Click the **General** tab to view and modify log and permissions settings.
  - Click the **Log** sub-tab to view any logged events that are triggered by the template object.

    **IMPORTANT** You must have the *Read* permission to view the Log tab.

    For more information about options found on the Log tab, see Viewing log events.

- On the **Permissions** sub-tab, you can configure the users or groups to whom you want to grant permissions to the new template object.

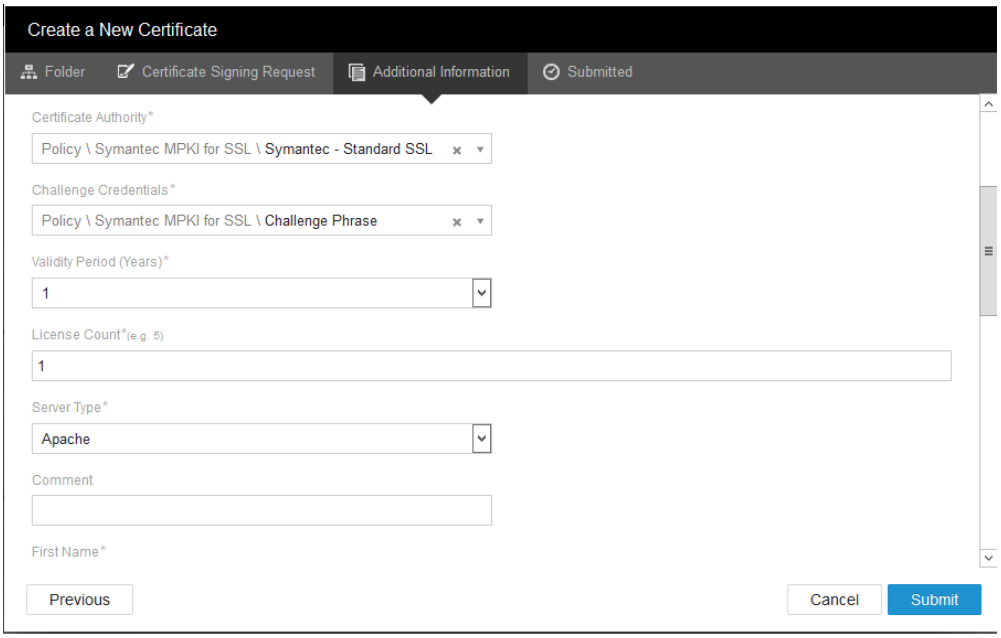
  Consider managing object permissions via parent objects so that you can take advantage of inheritance. For more information, see Permission inheritance and flow down.

## Symantec MPKI settings–certificate object settings

When you define a CA Template object, you provide the information Trust Protection Foundation needs to connect to the CA. However, when you associate the Symantec Managed PKI for SSL (Symantec MPKI) CA Template object configuration with a specific certificate object, you must define additional settings that are specific to the certificate–information such as the Validity Period, License Count, and Challenge Credential. These settings are passed to the Symantec MPKI CA when Trust Protection Foundation renews the certificate.

When you select the CA Template in the Certificate object configuration, you must complete the Symantec MPKI settings.

- The Symantec Managed PKI for SSL settings appear on the Additional Information tab of the Renewal Details or Create a New Certificate pages.



## Symantec MPKI settings in the Certificate Object Configuration

The following table reviews the Symantec Managed PKI for SSL settings in the Certificate object configuration.

| Field | Policy | Description |
|---|---|---|
| **Issuance Settings** | | |
| Challenge Credential |  | Password credential required to retrieve or revoke a Symantec certificate.<br><br>The Challenge Credential is given to Symantec when the certificate's CSR is submitted for enrollment. Symantec subsequently requires the Challenge Credential to retrieve or revoke the certificate.<br><br>**To select a challenge credential**<br><br>1. Click  to open the **Credential Selector** dialog.<br><br>2. Select the certificate credential you created with the Symantec MPKI administrator certificate, and then click **Select**.<br><br>For more information, see Managing system credentials in the *CyberArk Trust Protection Foundation Administration Guide*. |
| Certificate Validity<br><br>**TIP** The Certificate Validity option is available only when the **Allow Users to Specify End Date** option is enabled on the associated CA template. For more information, see "Allow Users to Specify End Date " on page 122. | No | Specify the period of time that the certificate is valid by selecting one of the following options:<br><br>■ **Use Validity Period**: Select this option (default setting) and then select a period of time (years) from the Validity Period drop-down list if you want to select a validity period specified on the associated CA |

| Field | Policy | Description |
|---|---|---|
| | | template. For more information, see "Validity Periods (Years)" on page 200.<br>■ **Specify End Date**: Select this option and then type an expiration date in the **End Date** field for certificates requested from the CA so that they don't expire during your known freeze periods.<br>If a specific certificate is being renewed or replaced, the Specify End Date is ignored. If a specific end date is required for a certificate that needs to be renewed or replaced, then create and enroll it using a new certificate object.<br>The options available in this menu are determined by the Available Validity Periods configured in the Symantec MPKI CA Template object configuration. For more information, see "Configuring the Symantec Managed PKI for SSL CA template" on page 196. |
| License Count |  | Number of licenses (i.e., concurrent installations) to be issued with the certificate. |
| Server Type |  | Type of server that the certificate is installed on. |
| Comment |  | Any information you want to store with the certificate. |
| **Symantec MPKI Owner** | | |
| First Name |  | Symantec MPKI Certificate Owner's first name. |

| Field | Policy | Description |
|---|---|---|
| Last Name |  | Symantec MPKI Certificate Owner's last name. |
| Email |  | Symantec MPKI Certificate Owner's email address. |
| Additional Certificate Fields | No | Your certificate might include additional certificate fields. These are custom fields defined by your organization in the Symantec MPKI Control Center. Trust Protection Foundation includes these values in the certificate signing request it submits to the Symantec CA.<br><br>If the custom fields are required, you must enter values for the fields in the Certificate object. If you do not complete required custom fields, the Symantec CA does not accept the CSR. If the fields are optional, you can leave the fields empty in the Certificate object. |

# VikingCloud

To enable Trust Protection Foundation to manage VikingCloud certificates, you must configure the VikingCloud CA Template object. This object provides the information that Trust Protection Foundation needs to request, retrieve, renew and revoke certificates issued by the VikingCloud CA.

## Configuring the VikingCloud CA template object

Enrollment of Organization Validated (OV) and Domain Validation (DV) certificates typically require an hour to complete from the time that they are submitted. However, the enrollment of Extended Validation (EV) certificates can take two to three days because the validation process is more rigorous.

Every 30 minutes, Trust Protection Foundation checks to see if the EV certificate is available. If a certificate cannot be retrieved after three consecutive days (the default setting), the certificate object is placed into an error state until an administrator attempts to retrieve the certificate again.

You can attempt to retrieve a certificate by using the **Retry** button on the certificate object.

**BEST PRACTICE** Consider managing CA Template object settings using a policy. For more information, see "Managing CA templates using policies" on page 58 in the *Certificate Authority and Hosting Platform Integration Guide*.

### To create and configure a VikingCloud CA template

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. From the **Tree** drop-down menu, click **Policy**.
3. In the **Policy** tree, select the folder where you want to create the CA Template object, and then click **Add**.
4. Click **CA Template**, then select **VikingCloud** to create it.
5. In the **CA Name** box, type a name for the new VikingCloud object.
6. Refer to the following table to complete the remaining CA template settings:

#### VikingCloud CA Template Configuration

| Field | Description |
|---|---|
| **Configuration** | |
| Credentials | The username and password credential obtained from VikingCloud that is used to authenticate with their CA web service for certificate enrollment and revocation. |
| Reseller ID | An identifier assigned by VikingCloud that uniquely identifies each customer's account. |
| Connect to test service | Configures the CA template to use VikingCloud's non-production web service and should be used for testing purposes only.<br>When this option is enabled, certificates enrolled using the CA template are generally *not* trusted by clients because they are issued by a test authority. |
| Validate | Press this button to validate the connection to the VikingCloud CA and to retrieve the certificate products that are available for enrollment. |

| Field | Description |
|---|---|
| **Certificate Fields** | |
| Product | The VikingCloud certificate product that will be requested when the CSR is submitted to the CA.<br><br>For VikingCloud, the certificate product defines the validity period of the certificate as well as the level of validation that is performed–OV, DV, or EV. |
| Subject Alt Name Enabled | Configures the current CA template object to support CSRs with DNS-based subject alt name (SAN) values.<br><br>**NOTE** DNS SANs are included in the Unified Communications Certificates that are used with Microsoft Exchange 2007 and Microsoft Office Communications Server.<br><br>If you do not select this option, the current CA template object will not accept CSRs with SAN values. If Trust Protection Foundation attempts to submit a CSR with SAN values, the CA Template object returns the following error:<br><br>Stage 400 (Creating CSR) SubjectAltName not supported by Application and CA.<br><br>**IMPORTANT** Before selecting this option, you must also verify the SAN feature is enabled on your CA engine. It it isn't, the CA returns the same error message when Trust Protection Foundation attempts to submit a CSR with SAN values.<br><br>For Secure Email (S/MIME) products, at least one Email SAN is required in the certificate request.<br><br>For additional information on defining SAN values in the certificate, see Subject Alt Name in the topic Certificate settings.<br><br>**NOTE** Different product types support different quantities of SANs. Please contact VikingCloud for more information. |

7. (Optional) To see additional attributes, review the settings on the **Support** tab.

8. Click **Save**.

## What's next?

After you create a CA object, you can select it from the Policy tree, and then view important information and manage various settings.

- Click the **General** tab to view and modify log and permissions settings.
  - Click the **Log** sub-tab to view any logged events that are triggered by the template object.

    **IMPORTANT** You must have the *Read* permission to view the Log tab.

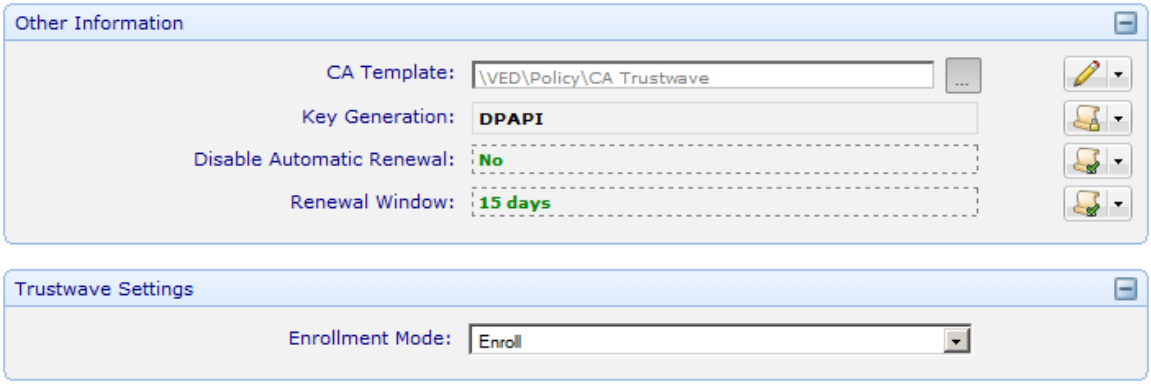
    For more information about options found on the Log tab, see Viewing log events.
  - On the **Permissions** sub-tab, you can configure the users or groups to whom you want to grant permissions to the new template object.

    Consider managing object permissions via parent objects so that you can take advantage of inheritance. For more information, see Permission inheritance and flow down.

## VikingCloud Settings

When you define a CA Template object, you provide the information Trust Protection Foundation needs to connect to the CA.

However, when you associate the VikingCloud CA Template object with a specific certificate object, Trust Protection Foundation presents an additional settings field for enrollment that applies to that certificate, specifically. Therefore, when you select the CA Template on the certificate object configuration page, you must select an enrollment mode in the **VikingCloud Settings** area.




The following table reviews the VikingCloud settings in the certificate object configuration.

| Field | Description |
| --- | --- |
| Enrollment Mode | Specifies the type of enrollment to use for the CSR being submitted to VikingCloud.<br><br>The Enrollment Mode list includes the following options:<br><br>■ **Enroll**: Select for new certificate enrollments.<br><br>■ **Renewal**: Select for renewing certificates.<br><br>■ **Reissue**: Select for re-issuance of existing certificates.<br><br>■ (Conditional) **ReplaceDV**: Available when configuring Domain Validated (DV) certificates. |

# UniCERT

To enable Trust Protection Foundation to manage UniCERT certificates, you must configure the UniCERT CA Template object. This object provides the information that Trust Protection Foundation needs to request, retrieve, renew and revoke certificates issued by the VikingCloud CA.

This chapter contains the following topics:

## Prerequisites configuration

Before you create the Verizon UniCERT CA Template object, you must first install the Verizon UniCERT connector–a Java service that enables Trust Protection Foundation to communicate with the Verizon UniCERT CA.

> **NOTE** You must log in with Administrator privileges to install the Verizon UniCERT connector package.

### To install the Verizon UniCERT connector package

1. Log in to the Windows server where Trust Protection Foundation is installed.
2. Navigate to the *drive*:\Program Files\Venafi\Packages directory.
3. From the Packages directory, launch the UniCERT connector package (Venafi UniCERT 5.3 Service x64.msi).

## Configuring the Verizon UniCERT CA Template object

To enable Trust Protection Foundation to manage Verizon UniCERT certificates, you must configure the Verizon UniCERT CA Template object. This object provides the information Trust Protection Foundation needs to request, retrieve, and install certificates issued by the Verizon UniCERT CA.

> **BEST PRACTICE** Consider managing CA Template object settings using a policy. For more information, see "Managing CA templates using policies" on page 58 in the *Certificate Authority and Hosting Platform Integration Guide*.

### To create and configure a Verizon UniCERT CA template

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. From the **Tree** drop-down menu, click **Policy**.
3. In the **Policy** tree, select the folder where you want to create the CA Template object, and then click **Add**.

4. Click **CA Template**, then select **UniCERT** to create it.

5. In the **CA Name** box, type a name for the new UniCERT object.

Refer to the following table to complete the remaining CA template settings:

### Verizon UniCERT CA Template Settings

| Field | Description |
|---|---|
| **Connection** | |
| Host | IP address or hostname to the UniCERT CA server.<br><br>Trust Protection Foundation supports both IPv4 or IPv6 connections. |
| Web Instance | Name of the current instance of the User Programmable Interface. |
| CA DN | Distinguished name of the UniCERT CA. This value is case-sensitive. |
| RA DN | Distinguished name of the registration authority (RA). This value is case-sensitive.<br><br>The RA is responsible for identifying and authenticating certificate subscribers before issuing certificates. |
| Credential | Certificate Credential required to connect to the UniCERT CA.<br><br>**To select the Credential**<br><br>Click the **Browse** button.<br><br>The Credential Selector dialog appears.<br><br>Select the Certificate Credential that stores certificate and private key password required to connect to the UniCERT CA server, and then click **Select**. |
| Validate | Tests the selected Certificate Credential to ensure Trust Protection Foundation can authenticate with the UniCERT CA server.<br><br>If the credential is valid and the connection succeeds, Trust Protection Foundation automatically populates the Policy Type drop-down list. |

| Field | Description |
|---|---|
| Port | Port that the UniCERT CA User Programmable Interface listens on.<br><br>The default port is 8088. For secure communication (SSL), the default port is 8443. |
| Secure Connection | Requires Trust Protection Foundation to connect to the UniCERT CA over a secure connection. (port 8443)<br><br>To use SSL with your UniCERT CA, the root certificate must be imported to the Java Keystore on the Trust Protection Foundation server. The command to import the root certificate to the Java Keystore is as follows:<br><br>"*java_pathto*\bin\keytool" -import -file *pathto*\ *root*.crt -keypass changeit -keystore "*install_drive*\Program Files\Venafi\Trust Protection Foundation\ Utilities\UpiProxy\rt\lib\security\cacerts" -storepass changeit |

| Field | Description |
|---|---|
| **Options** | |
| Policy Type | The UniCERT certificate policy you want to associate with the current CA Template object. Trust Protection Foundation references this policy when it submits the CSR.<br><br>A policy defines everything associated with the issuance of a particular type of certificate. The certificate that is issued from a policy request contains the default content with values derived from the parameters associated with those defaults.<br><br>For example, you could set up a profile for an SSL certificate that defines all aspects of that certificate including a default validity period of two years. When a user sends a request associated with this profile, the issued certificate contains the information specified in the profile defaults and is valid for two years.<br><br>If new Certificate Types are added to the CA, you must click **Validate** to update the Policy Type menu. |
| Subject Alt Name Enabled | Configures the current CA template object to support CSRs with DNS-based subject alt name (SAN) values.<br><br>**NOTE** DNS SANs are included in the Unified Communications Certificates that are used with Microsoft Exchange 2007 and Microsoft Office Communications Server.<br><br>If you do not select this option, the current CA template object will not accept CSRs with SAN values. If Trust Protection Foundation attempts to submit a CSR with SAN values, the CA Template object returns the following error:<br><br>Stage 400 (Creating CSR) SubjectAltName not supported by Application and CA.<br><br>**IMPORTANT** Before selecting this option, you must also verify the SAN feature is enabled on your CA engine. It it isn't, the CA returns the same error message when Trust Protection Foundation attempts to submit a CSR with SAN values.<br><br>For Secure Email (S/MIME) products, at least one Email SAN is required in the certificate request. |

| Field | Description |
| --- | --- |
| | For additional information on defining SAN values in the certificate, see Subject Alt Name in the topic Certificate settings. |

6. (Optional) To see additional attributes, review the settings on the **Support** tab.
7. Click **Save**.

### What's next?

After you create a CA object, you can select it from the Policy tree, and then view important information and manage various settings.

- Click the **General** tab to view and modify log and permissions settings.
  - Click the **Log** sub-tab to view any logged events that are triggered by the template object.

    **IMPORTANT** You must have the *Read* permission to view the Log tab.

    For more information about options found on the Log tab, see Viewing log events.
  - On the **Permissions** sub-tab, you can configure the users or groups to whom you want to grant permissions to the new template object.

    Consider managing object permissions via parent objects so that you can take advantage of inheritance. For more information, see Permission inheritance and flow down.

## ZTPKI CA

In order to use the ZTPKI CA CA driver, your account needs to be prepared by ZTPKI CA for automated access via the ZTPKI CA portal.

### To prepare your account for automated access

1. Obtain a Web Service Signing Certificate (Web API Certificate) issued by ZTPKI CA.

   This certificate is used by the driver to access the ZTPKI CA API.

2. Provide ZTPKI CA with the IP addresses of those Trust Protection Foundation servers that will be connecting to ZTPKI CA's APIs.

   For security reasons ZTPKI CA blocks API access from any address that is not registered. If NAT services are used on your network, you must register your external address.

3. Associate your Web Service Signing Certificate in ZTPKI CA with your IP address, and then configure firewall and web service access.

4. Using the ZTPKI CA web service, select which certificate templates can be requested and configure ZTPKI CA to enable these, accordingly.

   To protect your organization, only templates that you specifically request to be enabled can be used via the web service.

5. Make sure that a subscriber is registered to the account and assigned to the organizations that are going to request certificates through the web service.

The following topics include the steps required to set up the ZTPKI CA CA.

This chapter contains the following topics:

## Configuring the ZTPKI CA Template object

To enable Trust Protection Foundation to manage ZTPKI CA certificates, you must configure the ZTPKI CA CA Template object. This object provides the information that Trust Protection Foundation needs to request, retrieve, renew and revoke certificates issued by the ZTPKI CA CA.

### To create and configure a ZTPKI CA CA template

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. From the **Tree** drop-down menu, click **Policy**.
3. In the **Policy** tree, select the folder where you want to create the CA Template object, and then click **Add**.
4. Click **CA Template**, then select **ZTPKI CA** to create it.
5. In the **CA Name** box, type a name for the new ZTPKI CA object.
6. (Optional) To see additional attributes, review the settings on the **Support** tab.
7. Click **Save**.

Refer to the following table to complete the remaining CA template settings:

### ZTPKI CA CA Template Configuration

| Field | Description |
|---|---|
| **Connection** | |
| API Credential | The required API credential required to authenticate with the ZTPKI web service. |
| Environment | The geographic region and operational phase of your ZTPKI CA account. Select the endpoint that matches your account's location (US, EU, or AU) and intended use (Staging for testing, or Prod for issuing live certificates). |
| **Options** | |

| Field | Description |
|---|---|
| Template | The Policy Template for which certificates will be requested during enrollment. Policy Templates define Validity Period, Subject DN behavior, support for Subject Alternative Names (SANs), and the certificate type (e.g. Business SSL, EV SSL, etc.). |
| Validity Period | Specifies the amount of time for which the issued certificates will be valid. Select a duration, ranging from 1 day to 12 months. The available options are determined by the certificate profiles enabled on your ZTPKI portal. |
| Subject Alt Name Enabled | Configures the current CA template object to support CSRs with DNS-based subject alt name (SAN) values.<br><br>**NOTE** DNS SANs are included in the Unified Communications Certificates that are used with Microsoft Exchange 2007 and Microsoft Office Communications Server.<br><br>If you do not select this option, the current CA template object will not accept CSRs with SAN values. If Trust Protection Foundation attempts to submit a CSR with SAN values, the CA Template object returns the following error:<br><br>`Stage 400 (Creating CSR) SubjectAltName not supported by Application and CA.`<br><br>**IMPORTANT** Before selecting this option, you must also verify the SAN feature is enabled on your CA engine. It it isn't, the CA returns the same error message when Trust Protection Foundation attempts to submit a CSR with SAN values.<br><br>For Secure Email (S/MIME) products, at least one Email SAN is required in the certificate request.<br><br>For additional information on defining SAN values in the certificate, see Subject Alt Name in the topic Certificate settings. |

## What's next?

After you create a CA object, you can select it from the Policy tree, and then view important information and manage various settings.

- Click the **General** tab to view and modify log and permissions settings.
  - Click the **Log** sub-tab to view any logged events that are triggered by the template object.

    **IMPORTANT** You must have the *Read* permission to view the Log tab.

    For more information about options found on the Log tab, see Viewing log events.
  - On the **Permissions** sub-tab, you can configure the users or groups to whom you want to grant permissions to the new template object.

    Consider managing object permissions via parent objects so that you can take advantage of inheritance. For more information, see Permission inheritance and flow down.

The following table describes settings for the ZTPKI CA Template object.

### ZTPKI CA CA Template Object Configuration

| Field | Description |
|---|---|
| **Connection** | |
| API Credential | The required API credential required to authenticate with the ZTPKI web service. |
| Environment | The geographic region and operational phase of your ZTPKI CA account. Select the endpoint that matches your account's location (US, EU, or AU) and intended use (Staging for testing, or Prod for issuing live certificates). |
| **Options** | |

| Field | Description |
|---|---|
| Template | The Policy Template for which certificates will be requested during enrollment. Policy Templates define Validity Period, Subject DN behavior, support for Subject Alternative Names (SANs), and the certificate type (e.g. Business SSL, EV SSL, etc.). |
| Validity Period | Specifies the amount of time for which the issued certificates will be valid. Select a duration, ranging from 1 day to 12 months. The available options are determined by the certificate profiles enabled on your ZTPKI portal. |
| Subject Alt Name Enabled | Configures the current CA template object to support CSRs with DNS-based subject alt name (SAN) values.<br><br>**NOTE** DNS SANs are included in the Unified Communications Certificates that are used with Microsoft Exchange 2007 and Microsoft Office Communications Server.<br><br>If you do not select this option, the current CA template object will not accept CSRs with SAN values. If Trust Protection Foundation attempts to submit a CSR with SAN values, the CA Template object returns the following error:<br><br>`Stage 400 (Creating CSR) SubjectAltName not supported by Application and CA.`<br><br>**IMPORTANT** Before selecting this option, you must also verify the SAN feature is enabled on your CA engine. It it isn't, the CA returns the same error message when Trust Protection Foundation attempts to submit a CSR with SAN values.<br><br>For Secure Email (S/MIME) products, at least one Email SAN is required in the certificate request.<br><br>For additional information on defining SAN values in the certificate, see Subject Alt Name in the topic Certificate settings. |

## ZTPKI CA Settings—certificate settings

When you define a CA Template object, you provide the information that Trust Protection Foundation needs to connect to the CA as well as several other properties of the certificate to be enrolled. The ZTPKI CA template currently does not offer any settings that are specific to the certificate object to which it has been applied.

> **IMPORTANT:** ZTPKI CA may silently adjust your requested certificate values to comply with its template policies. This is a platform-wide behavior for many CAs. See "Understanding CA overrides and compliance" on page 51 in the Related Links below to learn why this happens and how to verify your final certificate details.

# 4

# Protecting server platforms and keystores

Application objects represent the server platforms or keystores where CyberArk Trust Protection Foundation™ manages certificates and private keys. Depending on the level of certificate management configured on the certificate–monitoring, enrollment, or provisioning–Trust Protection Foundation can request, enroll, renew, install, and validate the certificate on the application object's associated platform or keystore.

> **NOTE** For more information about monitoring, enrollment, or provisioning, see "About certificate lifecycle management" on page 42.

Trust Protection Foundation can also perform certificate and key management functions on managed applications, including generating, extracting, or pushing the private key associated with a managed certificate to the application object's associated platform or keystore.

> **NOTE** For more information about managing certificates on applications, see Associating Certificates with Applications in the *Trust Protection Foundation Certificate Management Guide*.

This section provides the information you need to configure and manage the platforms and keystores where managed certificates are located.

This chapter contains the following topics:

# Managing applications–overview

Application objects provide the configuration information Trust Protection Foundation needs to install and validate certificates on the Application object's associated platform or keystore.

To enable Trust Protection Foundation to provision certificates to supported platforms or keystores, you must complete the following:

1. Prepare the target system.

   This includes setting up SSL on the target system and granting the required system permissions. For a listing of supported applications and their associated appendix, see "CyberArk CA and application drivers library" on page 31.

2. (Conditional) If Trust Protection Foundation interacts with a device using an SSH command line and must use a jump server to access the device because it is behind a firewall, create a Jump Server object and configure its associated Device objects.

   For more information, see "Managing Jump Server Objects" on the next page.

3. In the Policy Tree, create a Device object for the network-accessible computer system where the certificate is installed.

   For more information, see "Managing device objects" on page 238.

4. In the Policy Tree, create and configure an Application object for the network appliance or keystore where the certificate is installed.

For more information, see "Managing application objects" on page 259.

5. In the Policy Tree, associate the Application object with the appropriate certificates.

   In the case of a network appliance, you associate each Application object with the relevant certificate consumed by the network appliance. For a keystore, you associate each Application object with the certificate stored in the keystore.

   > **TIP** Application objects can only be associated with a single certificate. However, a single certificate can be associated with many application objects.

   For more information, see Associating Certificates and Applications.

## Managing Jump Server Objects

A jump server is an intermediate server that external agents, such as CyberArk Trust Protection Foundation™, can use to access devices behind a firewall. When using a jump server to access devices, Trust Protection Foundation obtains the information required to access these devices from a jump server object.

The following sections provide the information you need to manage certificates via a jump server using Trust Protection Foundation.

This chapter contains the following topics:

## Jump server objects and lifecycle management

Trust Protection Foundation can both install and extract certificates and private keys through the jump server. Indeed, all system operations – except Network Validation which uses HTTPS – take place via the jump server. Trust Protection Foundation never accesses the jump server's associated application(s) directly.

> **NOTE** For more information about network and onboard validations, see "About certificate and application validation" on page 270.

When Trust Protection Foundation provisions certificates to servers that require a jump server connection, it securely copies the certificate and private key to the jump server. When the files are no longer needed, Trust Protection Foundation removes the files from the jump server.

> **NOTE** For more information on provisioning certificates, see How Provisioning Works.

In managing the jump server, itself, Trust Protection Foundation provides all standard logging and notification operations.

## Jump Server object hierarchy

The Policy tree provides a hierarchical view of your encryption deployment model, so Jump Server objects appear in context of other system objects such as Policy and Device objects.

In the Policy tree hierarchy, Jump Server objects are created under folder. Device objects can be created under Jump Server objects.

> **NOTE** Although you can create a Device object under its associated Jump Server object, this is not required, You can associate a Device object with any Jump Server object in the Policy tree.
>
> For more information, see "Managing device objects" on page 238.


## Jump server prerequisite configuration

Trust Protection Foundation supports jump server connections to the following supported applications using SSH:

| | |
|---|---|
| Apache | PEM |
| Cisco ACE | PKCS#12 |
| GSK | Tealeaf PCA |
| iPlanet | VAM nShield |
| JKS | |
| NetScaler | |

To support a jump server connection to a device behind a firewall, Trust Protection Foundation requires the following:

- Read/write access to a temporary folder on the jump server to transfer files to the target device
- SSH must be running on the jump server
- SCP and SSH must be in the jump server path
- Permissions to execute SSH and SCP commands on the jump server

**NOTE** Trust Protection Foundation does not support FTP communications with the jump server.

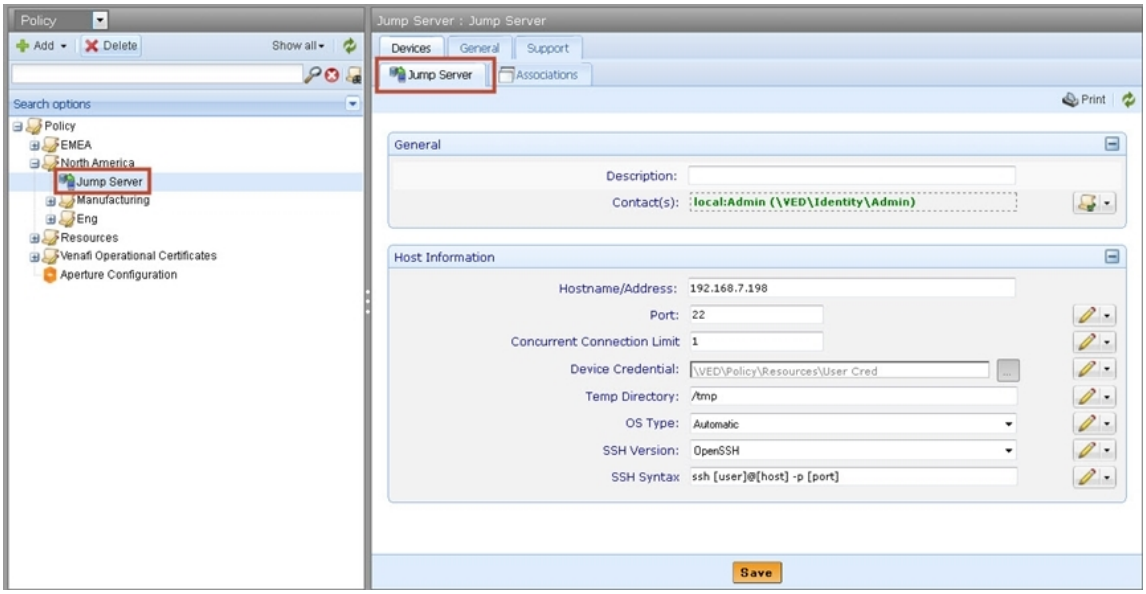
## Creating a Jump Server Object

You can create a jump server object in the Policy tree of Policy Tree. But before you begin, make sure that you have *create* permissions under the policy where you want to create the Jump Server object.

### To create a Jump Server object

1. From the Platform menu bar, click Policy Tree.
2. In the Policy tree, select the policy where you want to create the Jump Server object.
3. Click **Add** > **Jump Server**.

   The detail view displays the Jump Server object settings.




4. Define the Jump Server object settings, and then click **Apply/Save**.

   For a detailed description of the object settings, see "Jump server object settings" on the next page.

## Jump server object settings

To enable Trust Protection Foundation to use a jump server to access devices behind a firewall, you must configure the Jump Server object. This object provides the information Trust Protection Foundation needs to access devices from a jump server.

Additionally, you may want to consider managing the jump server object settings via policy. For more information, see "Managing jump server objects via policy" on page 236.

The following table describes settings for the Jump Server object. The second column indicates if the setting may also be managed via Policy.

| Field | Policy | Description |
|---|---|---|
| Jump Server | | |
| **General** | H | E |
| Jump Server Name | No | Unique name for the Jump Server object. |
| Description | No | Description for the Jump Server object. |
| Contact |  | User or group Identities assigned to this object. Default system notifications are sent to the contact identities.<br><br>The default contact is the master administrator.<br><br>**To select the Jump Server object contact(s)**<br><br>Click the **Browse** button.<br><br>The Identity Selector dialog opens. |

| Field | Policy | Description |
|---|---|---|
| | | If the Identity Selector dialog is not populated, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).<br><br>Select a User or Group Identity, and then click **Select**.<br><br>Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.<br><br>To configure this setting via Policy, go to the **Devices** > **Jump Server** tab in the Policy object configuration. The Jump Server Contact defined in the Policy object may be inherited by all subordinate Jump Server objects. |
| **Host Information** | | |
| Hostname/Address | No | IP address or hostname of the physical server associated with the Jump Server object.<br><br>Trust Protection Foundation supports both IPv4 or IPv6 connections. |
| Port |  | The port Trust Protection Foundation uses to communicate with the jump server.<br><br>Trust Protection Foundation uses the SSH protocol to communicate with jump servers. The default SSH port assignment is port 22.<br><br>To configure this setting via Policy, go to the **Devices** > **Jump Server** tab in the Policy object configuration. The Jump Server Port defined in the Policy object may be inherited by all subordinate Jump Server objects. |

| Field | Policy | Description |
|---|---|---|
| Concurrent Connection Limit |  | Maximum number of connections the server will accept from Trust Protection Foundation.<br><br>Currently, only one connection is supported.<br><br>To configure this setting via Policy, go to the **Devices** > **Jump Server** tab in the Policy object configuration. The Jump Server Concurrent Connection Limit defined in the Policy object may be inherited by all subordinate Jump Server objects. |
| Device Credential |  | Credential that Trust Protection Foundation uses to authenticate with the Jump Server.<br><br>Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential may be a username and password or a private key.<br><br>If you use a Username Credential to authenticate with the jump server, the user account must have Read and Write permissions to the Jump Server object's Temp directory.<br><br>**To select the Jump Server credential**<br><br>Click the **Browse** button.<br><br>The Credential Selector dialog appears.<br><br>Select the Credential required to authenticate with this device, and then click **Select**.<br><br>For more information, see Working with system credentials in the *CyberArk Trust Protection Foundation Administration Guide*. |

| Field | Policy | Description |
|---|---|---|
| | | To configure this setting via Policy, go to the **Devices** > **Jump Server** tab in the Policy object configuration, and use the **Jump Server Credentials** setting. The Jump Server Credentials defined in the Policy object may be inherited by all subordinate Jump Server objects. |
| Temp directory |  | Directory where Trust Protection Foundation can securely copy certificate and private key files to and from the jump server during operations.<br><br>When the files are no longer needed, Trust Protection Foundation removes the files from the Temp directory.<br><br>To configure this setting via Policy, go to the **Devices** > **Jump Server** tab in the Policy object configuration. The Jump Server Temp Directory defined in the Policy object may be inherited by all subordinate Jump Server objects. |
| SSH Version |  | The version of SSH that Trust Protection Foundation uses to communicate with the jump server during operations.<br><br>To configure this setting via Policy, go to the **Devices** > **Jump Server** tab in the Policy object configuration. The Jump Server SSH Version defined in the Policy object may be inherited by all subordinate Jump Server objects. |
| SSH Syntax |  | The syntax required to initialize an SSH session with the jump server. The standard syntax for each supported SSH version is as follows:<br><br>■ For Tectia 4:<br><br>ssh [*user*]@ [*host*] -p [*port*]<br><br>■ For Tectia 6: |

| Field | Policy | Description |
|---|---|---|
| | | ssh [*user*]@ [*host*] -p [*port*] -exclusive<br>■ For Open SSH:<br>ssh [*user*]@ [*host*] -p [*port*]<br>To configure this setting via Policy, go to the **Devices** > **Jump Server** tab in the Policy object configuration. The Jump Server SSH Syntax defined in the Policy object may be inherited by all subordinate Jump Server objects. |
| Associations Tab | No | The **Associations** tab lists all device objects associated with the current jump server object. Trust Protection Foundation uses the jump server to connect with the associated devices when performing management operations.<br>Trust Protection Foundation supports jump server connections to the following supported applications using SSH:<br>Apache, Cisco ACE, GSK, iPlanet, JKS, NetScaler, PEM, PKCS#12, Tealeaf PCA, VAM nShield<br>For information on associating a Device object with a jump server, see "Associating a device with a jump server" on the facing page. |

| Field | Policy | Description |
|---|---|---|
| Load | n/a | Queries all Device objects currently configured in the policy tree and returns a list of all Device objects associated with the current Jump Server object. |
| Export | n/a | Exports the list of associated Device objects to a CSV, tab-delimited, HTML, or XML file.<br><br>This option is available in Policy Tree only. In Policy Tree, you can copy and paste the information into Notepad or a spreadsheet application. |
| Refresh | n/a | Refreshes the current view to display the most recent events triggered by the current object. |
| General Tab | | |
| **Log** | n/a | Provides a view of all events triggered for the current object.<br><br>An administrator must have a minimum of the Read permission to view this tab.<br><br>For more information on the Log tab options, see Viewing log events. |
| **Permissions** | n/a | On the Permissions tab, you select the users or groups to whom you want to grant permissions to the current object. Then, you select which permissions you want the users or groups to have. You can also manage object permissions via parent objects, including the root Platform object or the Trust Protection Foundation server object (found in the Platforms tree).<br><br>If you configure Permissions in a parent object, those permissions are inherited by all subordinate objects. |

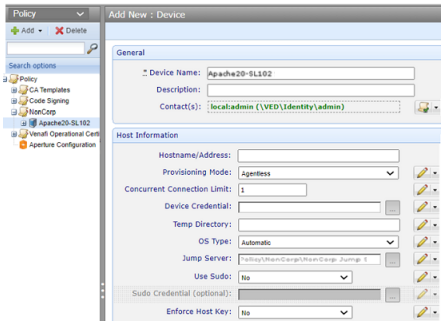
## Associating a device with a jump server

Device objects are assigned to jump servers in the Device object configuration.

**IMPORTANT** You must have the *View* and *Write* permissions to the device and jump server objects.

### To require a jump server connection for the current Device object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. In the Policy tree, select the Device object that requires a jump server connection.




3. In the Jump Server field, click the **Browse** button.
4. Select the Jump Server object required to connect to this device, and then click **Select**.

   When you create a Device object under a Jump Server object, Trust Protection Foundation automatically assigns the parent Jump Server object to the Device. However, this is not required. You can associate a Device object with any Jump Server object in the Policy tree.

## Managing jump server objects via policy

Policies allow administrators to define global configuration parameters, or “folders,” for encryption resources associated with that policy, including Jump Server objects. For example, within a policy, you can standardize the concurrent connection limit for all Jump Server objects created under that Policy object.

When a system object is defined under a Policy object in Policy Tree, the object is associated with the policy and is subject to the policy settings. For example, if you configured the Jump Server Host Information settings in a Policy object, Trust Protection Foundation would read those values for the policy’s subordinate Jump Server objects. Thus, policies can be used to standardize object configuration parameters and enforce security requirements.

You must have *write* permissions on the Policy object where you want to configure the Jump Server object settings.

For more detailed information on how folders work, see Using to Manage Encryption Assets in the *CyberArk Trust Protection Foundation Administration Guide*.

### To access the Jump Server settings in the Policy object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. In the Policy tree, select the Policy object where you want to configure Jump Server object settings, and then click the **Devices** > **Jump Server** tab.
3. In the **Detail View**, define the settings you want to apply to the policy's subordinate Jump Server objects.

   For details on each setting, refer to "Jump server object settings" on page 230.
4. Lock or unlock the value.

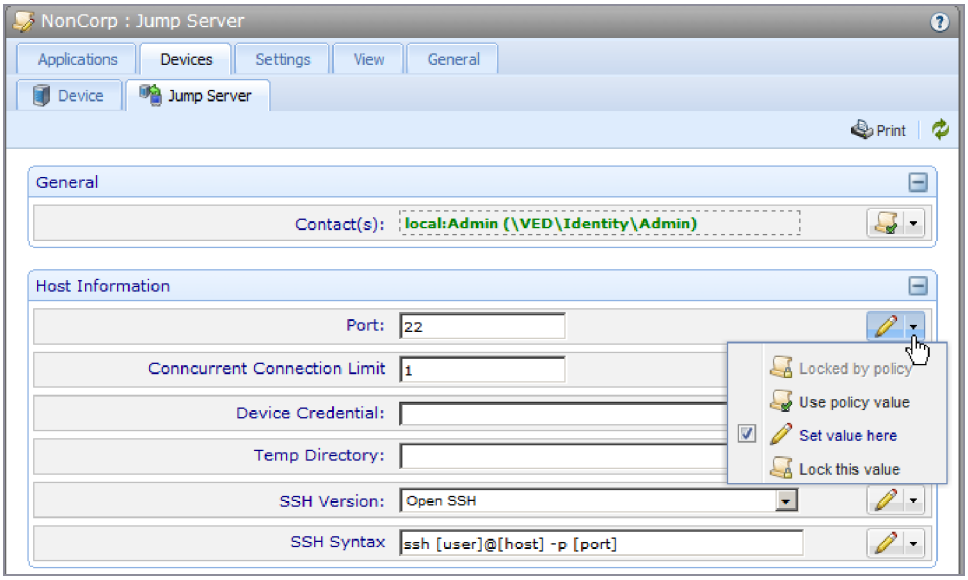
   When you define an attribute within a Policy object, you can lock or unlock the value. Locked attributes cannot be modified in objects contained by that policy, including sub-policies. Unlocked attributes function as suggested values; they can be accepted or overwritten in the subordinate object configuration.

   By default, policy values are unlocked  .

### To lock a policy value in Policy Tree

1. Click the attribute's drop-down menu .

   The menu shows the available options for the current attribute.




2. Click **Lock this value**.

   The attribute's icon changes to indicate the current value is in a locked state  .

# Managing device objects

Device objects represent the physical host on which certificates and private keys are installed. CyberArk Trust Protection Foundation™ references the device when it manages and validates certificates and private keys.

This chapter contains the following topics:

## Device objects and lifecycle management

For more information on provisioning certificates, see Certificate Installation (TrustForce). For more information about onboard validations, see "About certificate and application validation" on page 270.

Device objects may optionally be used in conjunction with Basic Application objects at the Monitoring or Enrollment level of management to associate devices with monitored certificates. This configuration allows administrators to track the servers where certificates are installed and route notifications to the Device contacts. For more information, see "Basic application configuration" on page 379.

## Device object hierarchy

- From the Platform menu bar, click Policy Tree.

The Policy tree provides a hierarchical view of your encryption deployment model, so Device objects appear in context of other system objects such as Policy, Application, and Certificate objects.

In the Policy tree hierarchy, Device objects are created under Policy or Device objects. Application object are created under Device objects. Certificate and Credential objects may be created under Device objects.

## Creating a device object in the Policy Tree

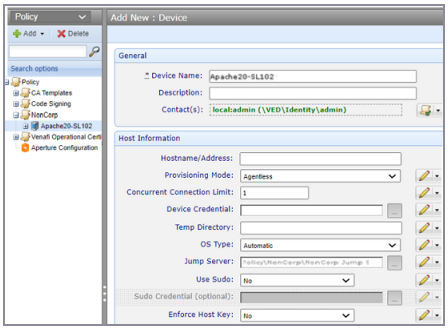
*Device objects* represent the physical host on which certificates are installed. Trust Protection Foundation references the device when it validates and manages certificates. Device objects are required to provision certificates and perform onboard validations. You must define a device object before you can create the associated application objects.

> **NOTE** You must have *create* permissions under the Policy or Device object where you want to create device objects.

### To create a device object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. In the **Policy** tree, select the policy or device object where you want to create the new device object.
3. Click **Add** > **Devices** > **Device**.

For a more information about the settings, see "Device object settings (agent, agentless, and adaptable provisioning)" below.




4. When you are finished, click **Save**.

## Device object settings (agent, agentless, and adaptable provisioning)

To enable Trust Protection Foundation to reference a device when it manages and validates certificates and private keys, you must configure the device object. This object provides the information Trust Protection Foundation needs to reference a device.

To get to these settings on an existing object, open Policy Tree, locate the device in the Policy tree and then click the **Settings** tab.

> **BEST PRACTICE** Consider managing device object settings via policy. For more information, see Managing devices using policies.

> **IMPORTANT** If you make any modifications to device object settings, you must click **Save** to implement those changes.

| Field | Policy | Description |
|---|---|---|
| **General** | | |
| Device Name | No | Unique name for the Device object. (Only visible when creating the device object.) |
| Description | No | Description of the device object.<br><br>**BEST PRACTICE** An effective description can be helpful to other administrators by describing the purpose or function of the device object. |

| Field | Policy | Description |
|---|---|---|
| Contact |  | User or group Identities assigned to this object. Default system notifications are sent to the contact identities.<br><br>The default contact is the master administrator.<br><br>**To select the Device object contacts**<br><br>1. Click the **Browse** button.<br><br>The Identity Selector dialog opens.<br><br>If the Identity Selector dialog is not populated, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups.If you want to display all user or group entries, enter the wildcard character (*).<br><br>2. Select a user or group identity, and then click **Select**.<br><br>3. Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups. |
| **Host Information** | | |
| Hostname/Address | No | IP address or hostname of the physical server associated with the device object.<br><br>Trust Protection Foundation supports both IPv4 or IPv6 connections. |

| Field | Policy | Description |
|---|---|---|
| | | **DID YOU KNOW?** Like most Trust Protection Foundation applications, AWS makes use of the Hostname/Address setting from its parent device object, even though it's a cloud service and not a typical device. This is because AWS is an Internet-based service with a public interface that is the same for all customers. But if you're provisioning to a secondary account, the parent device's Hostname/Address setting in Trust Protection Foundation is actually used to specify the AWS account ID (rather than the expected hostname or address).<br><br>This is a confusing but temporary method for managing this specific use case.<br><br>For more information about AWS configuration, see "Amazon Web Services (AWS)–Overview" on page 334. |
| Provisioning Mode |  | Specify whether provisioning to this device should be done using the Server Agent, or without the agent (called Agentless).<br><br>If you are provisioning using the Server Agent, you must also then configure agent work in Aperture. For more information, see Creating and assigning Work.<br><br>For Adaptable SSH Key Discovery, click **Adaptable**. |
| Concurrent Connection Limit |  | Maximum number of connections the server will accept from Trust Protection Foundation.<br><br>Currently, Trust Protection Foundation uses only one connection to provision certificates and keys on a server.<br><br>To configure this setting via Policy, go to the **Devices** > **Device** tab in the Policy object configuration.<br><br>The Device Concurrent Connection Limit defined in the Policy object may be inherited by all subordinate Device objects. |
| Device Credential |  | Credential that Trust Protection Foundation uses to authenticate with platforms or access keystores on the device. |

| Field | Policy | Description |
|---|---|---|
| | | Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential may be a password, a username and password, a certificate, or a private key.<br><br>Trust Protection Foundation uses the Device Credential only if credentials are not defined in the application object. If multiple applications are running on a single device and they share the same credentials, you can simplify credential management by defining the required credential only on the device object. The device credential is then used for all subordinate applications and you don't have to define credentials for each application object.<br><br>If you use the device credential to authenticate with your subordinate applications, the user account must have *read* and *write* permissions to the device object's Temp directory as well as the permissions required for each application.<br><br>For more information on application permissions requirements, refer to the Permission Requirements sections for each application driver.<br><br>**To select the Device Credential**<br><br>1. Click the **Browse** button.<br><br>2. In the Credential Selector dialog, select the Credential required to authenticate with this device, and then click **Select**.<br><br>For more information, see Working with system credentials in the *CyberArk Trust Protection Foundation Administration Guide*.<br><br>To configure this setting via Policy, go to the **Devices** > **Device** tab in the Policy object configuration, and use the **Device Credentials** setting.<br><br>The Device Credentials defined in the Policy object may be inherited by all subordinate Device objects. |
| Temp Directory |  | Directory where Trust Protection Foundation can write temporary files.<br><br>■ To configure this setting via Policy, go to the **Devices** > **Device** |

| Field | Policy | Description |
|---|---|---|
| | | tab in the Policy object configuration.<br><br>The Device Temp Directory defined in the policy object may be inherited by all subordinate Device objects.<br><br>For agentless discovery, the value of this field is used for .venafiTemp directories. (If a value is not in this field, agentless discovery will use the default directory of `/var/tmp`.) |
| OS Type | | Specify the OS type of the devices you will connect to with this policy, or select **Automatic**.<br><br>**TIP** This field is not visible if the *Provisioning Mode* is set to **Adaptable**. |
| Jump Server | | (Optional) The jump server required to connect with the current device.<br><br>A jump server is an intermediary server through which external servers, such as Trust Protection Foundation, can access a device behind a firewall. For more information, see "Managing Jump Server Objects" on page 226.<br><br>Trust Protection Foundation supports jump server connections to the following supported applications using SSH:<br><br>Apache, Cisco ACE, GSK, iPlanet, JKS, NetScaler, PEM, PKCS#12, Tealeaf PCA, VAM nShield<br><br>**To require a jump server connection for the current Device object**<br><br>1. Click the **Browse** button. |

| Field | Policy | Description |
|---|---|---|
| | | 2. In the Device Selector dialog, select the device object required to connect to this device, and then click **Select**.<br><br>When you create a Device object under a Device object, Trust Protection Foundation automatically assigns the parent Device object to the Device. However, this is not required, You can associate a Device object with any Device object in the Policy tree. |
| Use Sudo |  | Select **Yes** to enable sudo.<br><br>If you need to be able to provision certificates to secure locations for which the user account you're using does not have access rights, sudo can be used to temporarily elevate the user rights in order to perform provisioning tasks on a Unix server.<br><br>When enabled, Trust Protection Foundation provisions to a temporary directory where your user account has permissions to perform basic functions; then sudo is used to copy the files from the temporary directory to the target directory where they need to be placed.<br><br>For more information about sudo, see "About using sudo" on page 248.<br><br>**NOTE** Sudo use only applies to SSL CLI-drivers that use a standard shell and CLI-based drivers (Apache, JKS, GSK, iPlanet, PKCS#12, and PEM). |
| Sudo Credential (Optional) | | Select the credential to use with sudo, which in this case is simply a password.<br><br>This is the password that sudo asks for by default when a user attempts to run a command using sudo. The password is optional because the device administrator can put NOPASSWD in the sudoer's file, thereby allowing the user to execute the command without being challenged for a password.<br><br>For more information about sudo, see "About using sudo" on page 248. |
| Enforce Host Key |  | Describes how to handle SSH connectivity when the **Provisioning Mode** is *Agentless*. Matches the presented key and corresponding thumbprint with information in Trust Protection Foundation. |

| Field | Policy | Description |
|---|---|---|
| | | Set the **Enforce Host Key** to *No* (default), if you want host connectivity regardless of thumbprint changes. Trust Protection Foundation compares the current device thumbprint with the last seen fingerprint. If the fingerprints differ, log event is *40060020,SSH Public Key Fingerprint Changed*.<br><br>Set **Enforce Host Key** is *Yes*, if you want to block host connectivity after the thumbprint changes. If the presented thumbprint is different than the stored thumbprint, Trust Protection Foundation. refuses the connection. The log event is *40060004,SSH Connect To Host Failed*.<br><br>For example: *127.0.0.1, 2/22/2017 10:10:04 AM, \VED\Policy\centos-oracle: \VED\Policy\centos-oracle, Error: Error, Translated event: SSH Connection Failed, The SSH library failed to connect to 192.168.3.220 on port 22, with the Connection Result 8: The host key was not accepted.* |
| Presented Thumbprint | n/a | For devices that use the **Agentless mode** to provision. When the **Enforce Host Key** is *Yes*, this field shows the SSH server key thumbprint, also known as a fingerprint, that Trust Protection Foundation detected after a successful connection to the device. |
| Presented Key Type | n/a | The key type that encrypted the thumbprint. Corresponds to the **Presented Thumbprint** value. |

**TIP** The remaining settings in this list ONLY apply (and are thus only visible) if you set the *Provisioning Mode* to **Adaptable**.

## Adaptable Script Parameters

| | | |
|---|---|---|
| PowerShell Script |  | The PowerShell script you want used for this policy. The script must be stored in the `\Venafi\Scripts\AdaptableSSHManagement` folder on the Trust Protection Foundation server.<br><br>For more information about PowerShell scripts for Adaptable SSH Key Discovery, see PowerShell script reference for Adaptable SSH Key Discovery. |

## WebSDK OAuth Token Configuration

Adaptable SSH Key Discovery can use CyberArk APIs for communicating between devices. An API integration needs to be created for this purpose. To learn more about working with and configuring API integrations, see About API integrations.

<table>
<tr><th>Field</th><th>Policy</th><th>Description</th></tr>
<tr><td>OAuth Application Id</td><td>  </td><td>The value of the Client ID of the application from the API Integrations inventory.</td></tr>
<tr><td>OAuth Token Credential</td><td>  </td><td>The username credential object that contains the username and password of the identity that has access to the specified application. (This can be any SSH Manager for Machines identity that can obtain a grant for this application.)<br><br>When looking at the application in the API Integrations inventory, click on an application name, and look at the User or Team access section to see which users can obtain grants</td></tr>
<tr><td>OAuth Scope</td><td>  </td><td>This scope must exactly match the scope, privileges, and restrictions for this application as they are displayed in the API Integrations inventory. The easiest way to copy this is to open the application details from the API Integrations inventory, and in the Overview section, copy the value of the scope attribute in the JSON example.<br><br>At minimum, the application must have the following:<br><br><table><tr><th>Scope</th><th>Privileges and restrictions</th></tr><tr><td>ssh</td><td>discover</td></tr></table><br>If your application uses this minimum setting, you would enter this value:<br><br><code>ssh:discover</code><br><br>If you want to do SSH remediation on the devices, you will need the following minimum scope:<br><br><table><tr><th>Scope</th><th>Privileges and restrictions</th></tr><tr><td>ssh</td><td>manage,discover</td></tr></table><br>If your application uses this setting, you would enter this value:<br><br><code>ssh:manage,discover</code></td></tr>
</table>

**Advanced Settings**

| Field | Policy | Description |
|---|---|---|
| Enable Debug Logging |  | If you are trying to troubleshoot a problem with your Adaptable SSH Key Discovery setup, you may consider enabling this setting to get debugging information in the logs. |
| Script Execution Timeout |  | The maximum number of seconds a script can run before the process is terminated. |
| **Custom Fields** | | |
| | | If you have configured custom fields, you can set the field value on for the policy here. You will have a separate field for each custom field you have created. For more information, see Adding or deleting a custom field.<br><br>These custom fields are NOT passed to the PowerShell script, and are not related to the adaptable technology features of CyberArk Trust Protection Foundation - Self-Hosted. |

## What's next?

After you create a CA object, you can select it from the Policy tree, and then view important information and manage various settings.

- Click the **General** tab to view and modify log and permissions settings.
  - Click the **Log** sub-tab to view any logged events that are triggered by the template object.

    **IMPORTANT** You must have the *Read* permission to view the Log tab.

    For more information about options found on the Log tab, see Viewing log events.
  - On the **Permissions** sub-tab, you can configure the users or groups to whom you want to grant permissions to the new template object.

    Consider managing object permissions via parent objects so that you can take advantage of inheritance. For more information, see Permission inheritance and flow down.

## About using sudo

*Sudo* is a program for Unix-like computer operating systems that lets users run programs using the security privileges of another user (by default, the superuser).

Sudo lets you configure highly restricted user privileges on a remote device (refer to the commands listed in the table below). This new sudo option offers a more secure connection because root access is no longer required. Root access meant that if user account credentials were compromised, all other data on the remote device was potentially compromised, as well.

**DID YOU KNOW?** On Trust Protection Foundation, sudo is a method for provisioning certificates to a device using a non-root account and uses the default file ownership and permissions configured on your operating system. If your implementation requires different file ownership or permissions, Trust Protection Foundation drivers that support sudo include the ability to set them using a different feature of the driver.

When using sudo with Trust Protection Foundation, consider the following limitations and requirements:

- Workflow (SSH) Command Injection is not supported with sudo unless the injected commands are prefixed with **sudo** and no password is required by sudo to execute them.
- Only one certificate can be provisioned at a time to the same device when using sudo (the Concurrent Connection Count must be 1).
- The sudo account used by Trust Protection Foundation must have *read* and *write* permissions to the temp directory, as well as permissions to transfer files to and from the temp directory using SFTP.
- sudo can only be used with CyberArk drivers that make use of an SSH command-line interface and is only supported by platforms that use a standard, non-proprietary command shell.

| | Commands used with Central Generation | Commands used with Remote Generation |
|---|---|---|
| General Purpose | ls<br>rm<br>cp | ls<br>rm<br>cp |
| Set File/Owner Permissions | chmod<br>chown | chmod<br>chown |
| Apache and PEM | | openssl |

| | | |
|---|---|---|
| GSK | | gsk7cmd<br>gsk7capicmd<br>gsk8capicmd<br>gsk8capicmd_64<br>ikeycmd |
| iPlanet | certutil<br>pk12util | certutil<br>pk12util |
| JKS | | keytool |
| PKCS#12 | | (Not Applicable) |

When sudo is enabled, each command executed remotely on a device is prefixed with “sudo” to have the command execute in a privileged security context. The *sudoers* file governs which commands and which users are allowed. It can also specify whether or not the user must enter their password when prompted, an optional configuration that is supported by Trust Protection Foundation.

**EXAMPLE** In this example sudoer file, *venafi* is the user name that Trust Protection Foundation has been configured to use, */opt/pki* is the target directory, */tmp* is the temporary directory, and the commands are being executed via sudo without having to specify a password.

**IMPORTANT** The following example provides a general guide to get you started. Do not use the following example without replacing wildcard references with actual file path names wherever possible so as not to allow directory traversal.

```
# GSK, JKS, PEM, PKCS#12 central gen
venafi ALL= NOPASSWD:/bin/ls -ld /opt/pki*
venafi ALL= NOPASSWD:/bin/cp -pf /tmp/* /opt/pki/*
venafi ALL= NOPASSWD:/bin/cp /opt/pki/* /tmp/*
venafi ALL= NOPASSWD:/bin/cp -pf /opt/pki/* /opt/pki/*
venafi ALL= NOPASSWD:/bin/cp /tmp/* /tmp/*
venafi ALL= NOPASSWD:/bin/rm -rf /opt/pki/*.bak
venafi ALL= NOPASSWD:/bin/rm -rf /tmp/*
# for setting file owner/group and/or permissions
venafi ALL= NOPASSWD:/bin/chmod 0[0-7][0-7][0-7] /opt/pki/*
venafi ALL= NOPASSWD:/bin/chown * /opt/pki/*
venafi ALL= NOPASSWD:/bin/chmod 0[0-7][0-7][0-7] /tmp/*
venafi ALL= NOPASSWD:/bin/chown * /tmp/*
# iPlanet
venafi ALL= NOPASSWD:/bin/ls -ld /usr/bin/certutil, /bin/ls -ld /usr/bin/pk12util
venafi ALL= NOPASSWD:/usr/bin/certutil *, /usr/bin/pk12util *
venafi ALL= NOPASSWD:/bin/cp -pf /tmp/* /tmp/*
venafi ALL= NOPASSWD:/bin/ls -ld /tmp*
# PEM remote gen
venafi ALL= NOPASSWD:/usr/bin/openssl *
# JKS remote gen
venafi ALL= NOPASSWD:/usr/bin/keytool *
# GSK remote gen
venafi ALL= NOPASSWD:SETENV:/bin/sh -c ikeycmd *
venafi ALL= NOPASSWD:SETENV:/bin/sh -c gsk7cmd *
venafi ALL= NOPASSWD:SETENV:/bin/sh -c gsk7capicmd *
venafi ALL= NOPASSWD:SETENV:/bin/sh -c gsk8capicmd *
venafi ALL= NOPASSWD:SETENV:/bin/sh -c gsk8capicmd_64 *
```

For additional information and resources regarding sudo, visit the following websites:

- http://www.sudo.ws/
- https://www.garron.me/en/linux/visudo-command-sudoers-file-sudo-default-editor.html

# Managing devices

CyberArk Trust Protection Foundation - Self-Hosted includes the ability to manage devices that are already listed in the Device Inventory.

The Device Inventory uses the same filtering model as other inventories. The filters on the left are live filters; the results in the inventory list are updated automatically as you type. For more information see "Finding assets using filters on the Device Inventory list" on page 256.

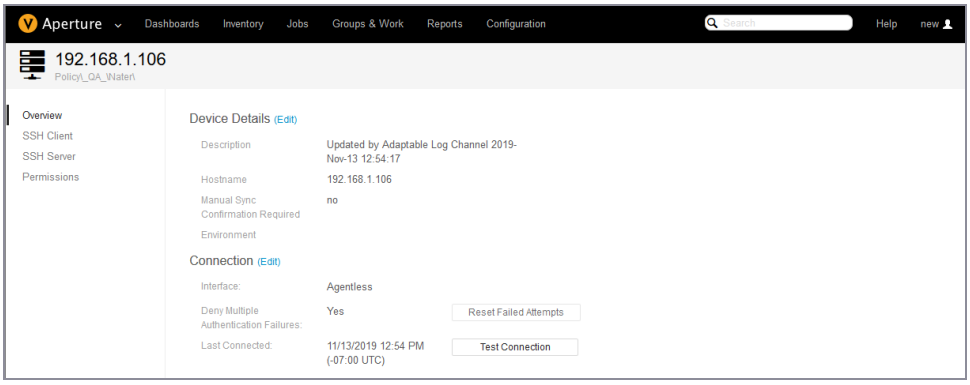
## Device Inventory list

In addition to viewing a list of, and being able to open the details of devices, there are two bulk actions you can take on the inventory page:

- **Scan for SSH keys.** Allows you to immediately connect to, and scan the selected devices for SSH keys.
- **Reset failed attempts.** Allows you to reset the SSH Scan Status for devices with a status of **Failed**. Devices show a status of failed when the "Deny Multiple Failures" option is set at the device or device policy level, and when Trust Protection Foundation has attempted to connect to the device and the key was rejected. If you want to reset failed attempts for all devices, you can click the reset button in the banner at the top of the screen. If you only want to reset failed attempts for specific devices, select the box next to the devices, then click **Reset Failed Attempts**.

| Field Name | Description |
|---|---|
| Name | Includes both the device name as well as the operating system of the device. If there is a value in parenthesis, this indicates the Environment set for the device. |
| SSH Trusts | Displays the number of incoming and outgoing SSH trusts established. (See Device Details, below, for more information.) |
| Interface | Indicates the interface used to interact with the device, if one exists. Options are *Agent*, *Agentless*, or *None*. |
| SSH Scan Status | Shows the results of the most recently-run SSH scan. If the status is *Pending*, the discovery is scheduled, but has not yet been started. If it doesn't start within 15 minutes, check that the *SSHManager* module is enabled on at least one engine that can process devices with *Pending SSH Scan* status. |
| Risks | Indicates risks identified for the device. |

## Device details screen




Click the links in the sidebar to view additional information about the device.

### Device Overview

The Device Overview tab shows the following information about the device:

- **Device Details**
  - **Description.** A free-text field used to describe information about the device.
  - **Hostname.** The URL or domain name of the device.
  - **Manual Sync Confirmation Required.** Whether a confirmation is required for manual synchronization for this device. In most cases, you'll leave this set to No. However, if you have a file server with stored key files for other devices, you can scan that device to acquire the keysets. However, these will require manual synchronization because Trust Protection Foundation isn't managing the actual synchronization between the server storing the keys and the target devices (for example, point of sale (POS) terminals, etc.).
  - **Environment.** A free-text field to describe this environment. Used for SSH keys, to identify risks if authorized keys and private keys are on devices in different environments. This can be set by policy at the folder level.
  - [Optional] ***Custom Fields.*** If you have specified custom fields for a device, they will be shown here.
- **Connection**

- **Interface.** Whether the device is connected via Agent or agentless connection. If the interface type is not *Agent*, and if the credential is connected to the device, you can edit the credential. If a credential doesn't exist, you can select or create a credential to associate with the device.

  For more information see Configuring connection settings for Agentless SSH devices

- **Deny Multiple Authentication Failures.** If Trust Protection Foundation is unable to connect to a device because the connection failed, You can prevent it from retrying. This enables you to prevent locking your account because of multiple failure attempts. For more information, see SSH policy settings details

- **Reset Failed Attempts.** If you had the Deny Multiple Authentication Failures feature enabled, and Trust Protection Foundation attempted to connect to the device and was rejected, you can click this button to have Trust Protection Foundation begin attempting to connect to the device again.

- **Test Connection.** This button allows you to have Trust Protection Foundation connect to the device, allowing you to verify that the current credential is valid.

Click the edit links to edit details about the device.

## SSH Client

The SSH Client tab has two sections:

- **Outgoing Access:** A server may be configured to connect to another machine using keys for authentication. In this section, you see every server account that this device can connect to using SSH keys. The fields on this screen are:

| Field | Description |
|---|---|
| Server/Account | The servers or accounts this device can access using the keys discovered on this device. |
| Client Account | The client account name used to make the connection to the external device. |
| Fingerprint | The fingerprint of the key used to authenticate with the external device. |
| Risks | Highlights known security risks associated with this connection. |

- **Trusted Servers:** A list of servers that are trusted on the device. Click the name of the server to see the device. Click the fingerprint to see the keyset.

| Field | Description |
|---|---|
| Server | The servers that are known on this device (based on known host key entry).<br><br>If more than three servers are listed for a single client account, only the first three are shown, but an indicator shows there are more.<br><br>Click the server name to see that device in the inventory. |
| Client Account | The client account name used to make the connection to the server.<br><br>The known host key's file path is shown under the account name. |
| Fingerprint | The fingerprint of the server's host key.<br><br>Click the fingerprint to see the keyset. |
| Risks | Highlights known security risks associated with this server trust. |

## SSH Server

The SSH Server tab has two sections:

- **Authorized Clients:** The device maintains a list of devices that are allowed to connect to it. The authorized clients list every client account that can connect to this device using SSH keys. The fields on this screen are:

| Field | Description |
|---|---|
| Client/Account | The information about the client device that is connecting to this device. |
| Server Account | The server account information the client device uses to authenticate to this device. |
| Fingerprint | The fingerprint of the key that the client device uses to authenticate to this device. |
| Last Used | The last time the client device tried to use this credential to connect to this device. |
| Risks | The known security risks associated with this connection. |

- **Host Keysets:** A list of host keysets of SSH server on this device. Click the name of the host keyset to see information about the keyset.

## Permissions

The permissions gives you detailed information about what users and groups have permissions to the device. You can add users or groups, and specify what permissions they should have, and you can see the cumulative permissions that are explicitly granted to a user or a group at the policy level higher up in the folder structure. You can also use the Troubleshoot Permissions link to see all permissions to the device granted to one or more specific users, including their group memberships.

For more information about how permissions work in Trust Protection Foundation, and how to use the Permissions tab, see About permissions in Aperture

## Finding assets using filters on the Device Inventory list

You can use filters to quickly find items in CyberArk Trust Protection Foundation - Self-Hosted inventories. Items that can be filtered include certificates, SSH keys, devices, identities, credentials, or Server Agents.

From any inventory list you can apply one or more filters to narrow the results. For example, use filters when you want to find a specific item, or find a group of items that meet a more specific set of criteria.

### Filter Panel Types

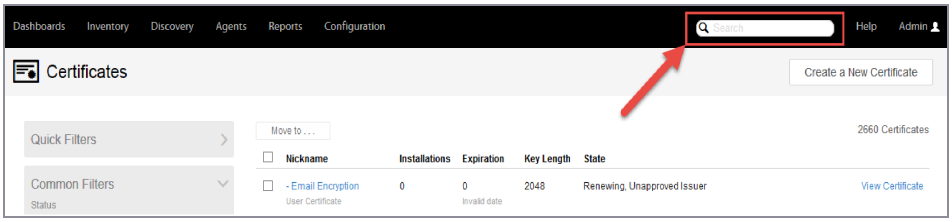
The filter panel contains the following filters:

- **Device Properties.** Unlike other inventory filters, the Device Inventory lists each filter in its own group. Those individual filters are described below.

### To search for Objects using quick search

> **NOTE** Quick Search does not allow you to search for agents.

1. In the **Search** box on the menu bar, type all or part of the name of the object you are looking for, and then press the Enter key.




2. In the search results, click the name of an object to view its details.

### To refine a list of discovered objects Using Filters

1. From the menu, open any of the inventory list views.

   For example, click **Inventory** > **Devices**.

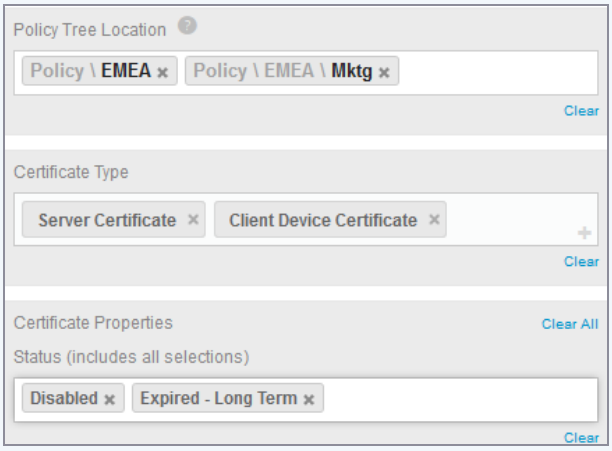
2. Using Filters, select and apply one or more filters to narrow the list of discovered items.

   > **DID YOU KNOW?** As you select and remove filters, the inventory list is automatically refreshed giving you instant filter results.

3. When you find the object you want in the inventory, click its name to view details.

**EXAMPLE** How search filtering works

All of the selections in a filter field are **OR** fields.




In the example above, the search could be described in the following way:

Show me all certificates with (policy location of EMEA or EMEA/Marketing) *and* (a certificate type of Server Certificate *or* a Client Device Certificate) *and* (Status of Disabled *and* Expired-Long Term).

## List of Device Properties filters

The following filters are available on the Credential Inventory page.

| Filter | Description | Multi-value Support | Type |
|---|---|---|---|
| Name | Filters based on the device name, host or IP address. | Yes / OR | Partial match search (starts with) |
| SSH Type | Filters based on the device type, whether it is a Client, or a Server. Client devices have outgoing SSH Trusts–they initiate the connection to the server. Server devices have incoming SSH Trusts–they act upon requests for connection from client devices. | No | Select from list |
| Folder | Filters based on the device's parent folder. A checkbox allows you to search through all sub-folders as well. | Yes / OR | Search from list |
| Status | Filters on the SSH scan status, allowing you to specify devices whose most-recent scan was complete, failed, or has never scanned. | Yes / OR | Search from list |
| Properties | Filters on one or more available properties, including Risks.<br>**NOTE** You will only see properties, including Risks, that exist in your data. | Yes / OR | Select from list |
| Custom Field | If you have created custom fields in your system, you can filter by them using this filter object. Select a custom field to add the filter criteria for that field. | No | Select from list |
| Discovered by | The name of the discovery job that discovered the device. | No | Select from list |
| Environment | Filter by the data set in the Environment field, whether that is set at the device or policy level. | Yes / OR | Partial match search (starts with) |

# Managing application objects

Application objects represent the server platforms or keystores where CyberArk Trust Protection Foundation™ manages certificates and private keys. Depending on the level of certificate management configured on the certificate–Monitoring, Enrollment, or Provisioning–Trust Protection Foundation can request, enroll, renew, install, and validate the certificate or perform key management functions on the Application object's associated platform or keystore.

The following sections provide the information you need to create and manage Application objects using Policy Tree. In Aperture, you can create the same objects by adding one or more installations to a certificate. For more information, see Working with certificate installations.

This chapter contains the following topics:

## Application Objects and Lifecycle Management

Note that only the Enrollment and Provisioning management levels are listed in the table because these are the only levels at which Trust Protection Foundation manages certificates. Trust Protection Foundation does not act on certificates or applications at the Monitoring level.

Individual stages may vary per application. For information on the certificate lifecycle stages for each application, see the application appendices in the *CyberArk Trust Protection Foundation Application Guide*.

## Managed stages of the certificate lifecycle

The following table outlines the managed stages of the certificate lifecycle.

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| Stages 0-700 are performed by the Application driver only if remote key generation is enabled. If the private key and CSR are locally generated on the Trust Protection Foundation server, stages 0-700 are performed by the X509Certificate Application driver.<br><br>The private key and CSR are remotely generated on the certificate's consumer application(s) if the **Generate Key/CSR on Application** option is enabled in the Certificate object. | | | | |
| 0 | StartProcessing | Trust Protection Foundation prepares the certificate for lifecycle processing. | x | x |
| 100 | CheckStore | Only applies to remote generations. | platform dependent | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| | | If the private key and CSR is generated remotely, Trust Protection Foundation compares the keystore or Directory configuration parameters specified in the Application object with the actual configuration on the application. | | |
| 200 | CreateConfigureStore | Only applies to remote generations.<br><br>If the certificate keystore does not exist, Trust Protection Foundation creates the keystore as per the configuration parameters defined in the Application object. | | x |
| 300 | CreateKey | Trust Protection Foundation creates the private key. | x | x |
| 400 | CreateCSR | Trust Protection Foundation creates the Certificate Signing Request (CSR). | x | x |
| 500 | PostCSR | Trust Protection Foundation submits the CSR to the Certificate Authority (CA). | x | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| | | If you post a manual CSR, this is the first stage of the certificate lifecycle. | | |
| 600 | ApproveRequest | Trust Protection Foundation approves the certificate renewal at the CA. | x | x |
| 700 | RetrieveCertificate | Trust Protection Foundation retrieves the certificate from the CA. | x | x |
| 800 | InstallCertificate | Trust Protection Foundation installs the certificate on the target application. | | x |
| 900 | CheckConfiguration | Trust Protection Foundation verifies the configuration after the certificate is installed. | | |
| 1000 | ConfigureApplication | Trust Protection Foundation configures the application to use the installed certificate, if needed. | | |

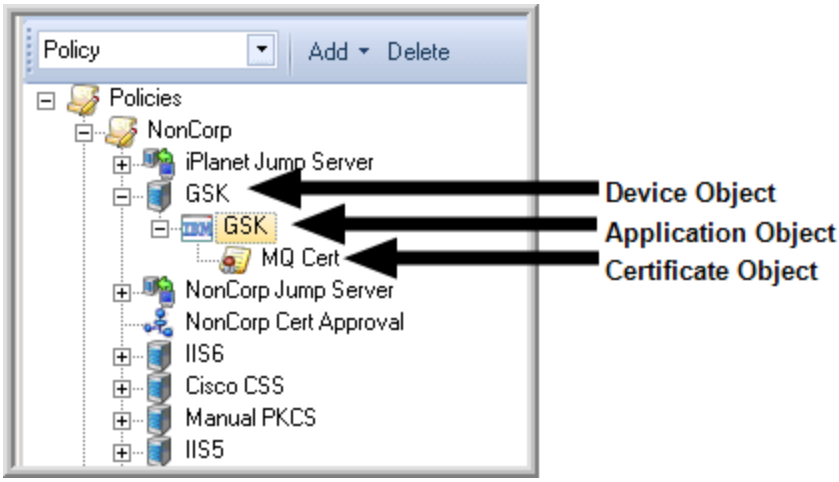
| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 1100 | RestartApplication | Trust Protection Foundation restarts the application after the certificate is installed and configured, if needed. | | |
| 1200 | EndProcessing | Trust Protection Foundation completes the certificate processing and, if configured, runs a Validation check on the certificate and application. | | x |
| 1400 | Revocation | Trust Protection Foundation submits a revocation request to the CA.<br>Certificate revocation is a certificate operation; it does not involve the application driver. | x | x |

## Application object hierarchy

- From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.

The Policy tree provides a hierarchical view of your encryption deployment model, so Application objects appear in context of other system objects such as Policy, Device, and Certificate objects.

In the Policy tree hierarchy, Application objects are created under Device objects. Optionally, certificate objects can be created under Application objects.



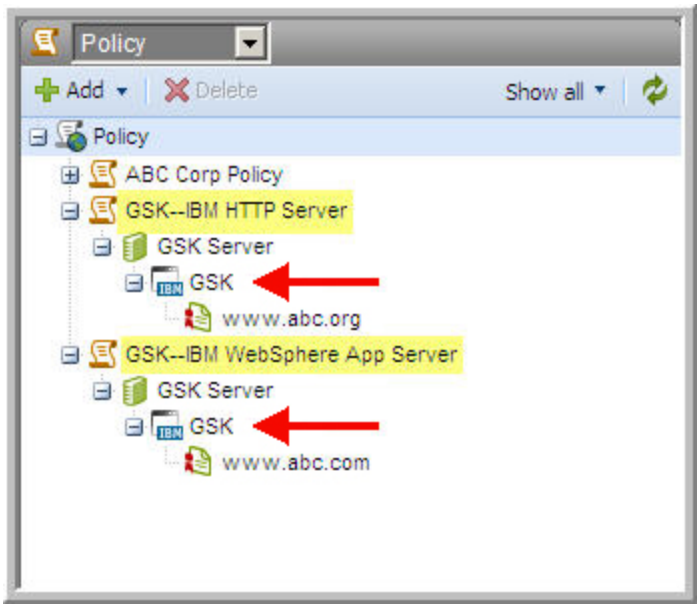

## Managing multi-object applications

For example, if a GSK Application object maintains a CMS keystore that stores certificates consumed by multiple platforms—e.g., the IBM WebSphere HTTP Server and the IBM WebSphere Application Server—you can create a separate policy for each consumer platform. In this case, you might create a policy for certificates consumed by the IBM HTTP Server and another policy for certificates consumed by the IBM WebSphere Application Server.

To apply the platform folders, you must create a separate GSK Application object under each policy, and then associate each GSK Application object with its respective certificates.

In the end, the GSK Application object configurations are identical—they are just created under different folders and associated with different certificates.

Figure 4.1: Multiple Application Objects for a Single Keystore




## Creating an application

*Application* objects represent the server platforms or keystores that use certificates to provide TLS connections for secure communications. They also represent installations of certificates.

When you create an application, you provide all of the configuration information Trust Protection Foundation needs to manage certificates for your chosen platform or keystore. Depending on the application, this may include certificate paths and filenames, application credentials, private key credentials, and so forth.

**NOTE** You must have the *Create* permission on the *device* where you want to create the application.

Application objects can only be created under device objects.

Device objects represent the physical host on which certificates and private keys are installed.

### To create an application object

**TIP** It's a good idea to create the prerequisite object first so that credentials are available to select when you create the application object.

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select an application object template.
3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.
4. (Optional) In the **Device Certificate** box, click  to select and associate a certificate with the new application.

   > **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

   For detailed information on associating a certificate with an application, see Associating a certificate with an application object.
5. Under **General**, do the following:
   a. In the **Application Name** field, type a name for the new application.
   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.
   c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

      Default system notifications are sent to the contact identities. The default contact is the master administrator.

> **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
>
> Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

   The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

   You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

   For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Under **Application Information**, do the following:

   a. Click  next to **Application Credential** to browse for the credential object that you want to use to authenticate with the application.

   > **DID YOU KNOW?** Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential might be a user name or private key credential; some drivers–such as F5, which is not SSH-based–can only use the user name credential for authentication.

   > **NOTE** The user account you select must have *Read* and *Write* access to the Temporary, Private Key, and Certificate directories.
   >
   > If you need help with this step, see your system administrator.
   >
   > For more information, see Working with system credentials in the *Administration Guide*.

> **DID YOU KNOW?** The **Connection Method** is the protocol that Trust Protection Foundation uses to connect to the server and manage the certificates installed on that server. In an application object's settings, this field is typically read-only.

b. (Conditional) In the **SSH Port** field, specify the port number that Trust Protection Foundation should use to communicate with the appliance via an SSH connection.

   The default SSH port assignment is 22.

c. In the WinRM Port, specify the port number that Trust Protection Foundation should use to communicate with CAPI devices.

   The default WinRM ports assignments are port 5985 and 5986 for HTTP and HTTPS, respectively.

d. (Optional) In the **Port** field, type the port that Trust Protection Foundation should use to communicate with the server where the application is installed.

   Trust Protection Foundation uses the SSH protocol to communicate with the application server installed on Linux or Windows. The default SSH port assignment is port 22.

e. From the **API Protocol** drop-down, select either **SSL** or **TLS**.

   This value must match the Secure+ Protocol configured under the Security Options of the Connect:Direct endpoint to which Trust Protection Foundation is provisioning.

f. (Optional) In the **Port** field, type the TCP port on which the Connect:Direct endpoint is listening for API connections.

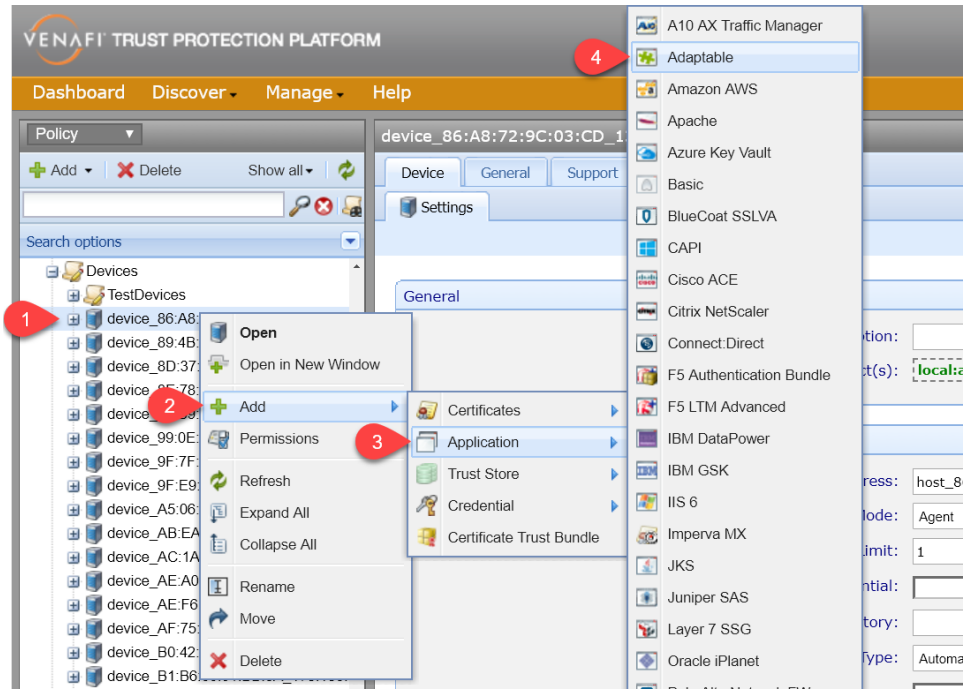
   The default API port assignment is 1363.

9. Click the application you want to create and then complete the new application's settings.

   Application settings vary depending on the associated platform or keystore requirements.

   Refer to the "Protecting server platforms and keystores" on page 223 sections of the documentation for your specific platform or keystore.

   If you need help managing applications, contact your System Administrator.




10. When you are finished, click **Save**.

## Managing applications using policies

In Trust Protection Foundation, a folder is a policy. Folders let administrators define global configuration parameters for assets associated with a policy, including CA template, device, and application objects.

When an object is defined under a folder (policy), that object is then associated with the folder and is subject to that folder's settings. Trust Protection Foundation reads values for the folder's subordinate objects. For example, if you configured the Device Host Information settings in a folder object, Trust Protection Foundation would read those values for the policy's subordinate device objects. Thus, policies can be used to standardize object configuration parameters and enforce security requirements.

Thus, policies can be used to standardize object configuration parameters and enforce security requirements.

For more detailed information on how folders work, see Using policies to manage encryption assets in the *CyberArk Trust Protection Foundation Administration Guide*.

> **NOTE** You must have *write* permissions on the Policy object where you want to configure the CA Template object settings.

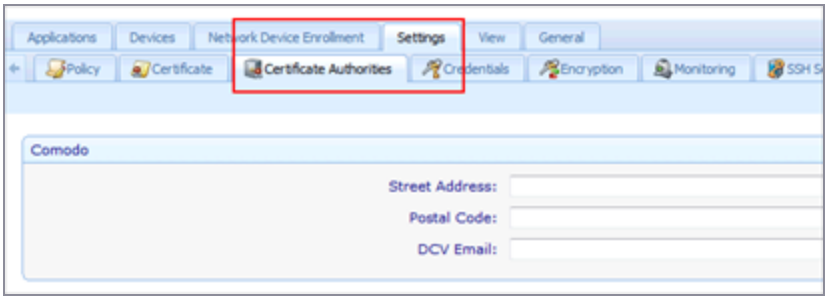
### To access application object settings in a policy object

1. From the Platform menu bar, click Policy Tree.

   You must have *write* permission on the Policy object where you want to configure the CA Template object settings.

2. In the Policy tree, select **CA Templates**.

3. In the **Settings** tab, click **Certificate Authorities.**




   The Detail View shows all of the settings that are configurable for each CA template.

4. Find the CA template you want, then define the settings you want to apply to the policy's subordinate CA template objects.

5. Lock or unlock the value of each attribute.

   *Locked* attributes cannot be modified in objects contained in that policy, including sub-folders. *Unlocked* attributes are suggested values. They can be accepted or overwritten in sub-folders.

   By default, policy values are unlocked  .

## About certificate and application validation

The CyberArk Trust Protection Foundation™ validation feature validates your encryption system certificates and applications to determine if the correct certificate is being used. It also determines if it is functional and if it is installed in the correct place on its associated platform or keystore.

Security, compliance, and technological innovation have introduced new criteria for the validation of certificates and the servers that host them.

Trust Protection Foundation offers certificate validation that is easy to use, provides more actionable results, helps assure compliance with industry standards, and supports best security practices.

- You can configure validation and see results in Certificate Manager - Self-Hosted.
- When you view validation results, you can clearly see every target that was scanned for the certificate.
- Validation provides complete, qualitative analysis of the certificate chain presented by the hosting server.
- Validation identifies the SSL or TLS protocols that the hosting server supports. CyberArk provides these communication protocols: SSL 2.0, SSL 3.0, TLS 1.0, TLS 1.1, TLS 1.2, TLS 1.3.
- Different certificates that are hosted on the same IP address and port can be validated.

For more information about configuring application and certificate validation, see SSL/TLS network validation in the *Certificate Management Guide*.

## Managing applications using HSM-protected keys and Advanced Key Protect

Hardware security modules (HSMs) offer a way to provide a higher level of security by generating private keys within a protected domain from which they're never permitted to leave. This means that an attacker would have to do all traffic decryption work from the application server they compromised, which would then put themselves at much greater risk of being detected.

Interfacing an application server with an HSM requires additional software, configuration, and expense.

HSMs are generally reserved for protecting the most mission-critical applications and those that manage highly sensitive data. CyberArk can automatically process generating keys inside Thales SafeNet Luna SA and Entrust nShield HSMs using the following provisioning drivers:

- Apache
- CAPI
- JKS

Using remote key generation via an application driver to an HSM allows the HSM to complete the full certificate lifecycle. This remote generation requires that target devices have the necessary HSM client software installed and properly configured.

**EXAMPLE** The following diagram provides an overview of how you configure your HSM using Trust Protection Foundation, an application server (running Apache, Java, or IIS), your HSM client utility and HSM server, and your CA.

1. Execute command to generate keypair inside the HSM by Trust Protection Foundation using your HSM's client utilities and is remotely executed from the Apache/Java/IIS host (the Application server).
2. Execute command to generate CSR on the application server (Apache/Java/IIS host) using your HSM's client utilities and then executed, retrieved, and submitted by Trust Protection Foundation to the CA.
3. A certificate is issued by the CA and then retrieved and provisioned by Trust Protection Foundation to the application server (Apache/Java/IIS host).

**IMPORTANT** If you want to configure remote key generation using an HSM, you must activate Advanced Key Protect, an optional add-on feature to CyberArk Trust Protection Foundation. For more information, see Advanced Key Protect in the *Trust Protection Foundation Administration Guide*.

Normally remote generation only supports one certificate to one application. If you want to install a certificate to many locations, the HSM can do central generation. However, if you are an Advanced Key Protect customer and have HSM connected Apache installations, we do support installing a single certificate to many Apache servers and making sure the Apache is configured to access the private key on the HSM properly. For special configuration information, see "Configuring HSM-based remote key generation" on the facing page.

**DID YOU KNOW?** In asymmetric encryption, security relies upon private keys remaining private. Limiting access to private keys is essential to ensuring that network communications between client and server are secure and protected. This is typically done by setting permissions on keystore files at the operating system level of the computer hosting the certificate and by applying password-based encryption on the keystore itself.

Although this method protects the private key from unauthorized access, it does not prevent a privileged user (or an attacker who has stolen a privileged user's credentials) from exfiltrating the keystore from the computer and using readily available password-cracking methods to obtain the private key. The possessor of a stolen private key can decrypt communications between the client and server putting sensitive data at risk.

## Configuring HSM-based remote key generation

When planning to set up HSM-based remote key generation, review the following important configuration steps:

1. You must enable Advanced Key Protect, an optional feature in the CyberArk Configuration Console.

   For more information, see Advanced Key Protect in the Administration guide.

2. On the certificate being provisioned, enable remote key generation by setting **Generate Key/CSR on Application** to **Yes**.

   **IMPORTANT** When one or more of the following items are true, Remote Generation is *not* supported and, therefore, the *Generate Key/CSR on Application* setting is ignored.

   - You're using a driver that does not support remote generation.

     To learn which drivers support remote generation, see "Supported integrations: devices, applications, services and features supported by CyberArk" on page 40.

   - You're using a self-signed CA template to enroll a certificate; self-signed CA templates do not work with remote generation.

     This is because Trust Protection Foundation requires that the private key be stored centrally so that it can be used to sign the self-signed certificate.

   - Your certificate is associated with more than one application; to work correctly, the certificate must be associated with *one* application.

   - You have not set the certificate's management type setting to Provisioning.

3. (Recommended) Apply the **Reuse Private Key for Service Generated CSRs** policy to the folder containing the certificate by setting it to **Yes**.

   **BEST PRACTICE** Although it is generally a best practice to generate a new key pair with every certificate renewal, the strong key protection of the HSM makes key re-usage acceptable. It prevents multiple generations of keys from accumulating in the HSM over time since deletion of a certificate from the device generally does not also remove keys from the HSM.

4. (Conditional) To complete the configuration, refer to the related topic and to the associated vendor documentation for configuration details specific to the driver you're using:

- **Apache:**
  - If you want to use Aperture to create an installation or a group of Apache installation applications that share the same certificate, see "Creating an Apache Installation" on page 355.
  - "Creating an Apache application object" on page 357
  - Vendor documentation:

    Apache HTTP Server Integration Guide (007-011228-001)

    Entrust nShield Apache HTTP Server Integration Guide
- **CAPI:**
  - "Creating a CAPI application object" on page 390
  - Vendor documentation:

    Microsoft Internet Information Services Integration Guide (007-011955-001, Rev. N [Aug 2018])

    Entrust nShield e-Security Integration Guide for Microsoft IIS 8.0 Windows Server 2012 and Windows Server 2012 R2
- **JKS:**
  - "Creating a JKS application object" on page 510
  - Vendor documentation:

    Oracle WebLogic Server Integration Guide (007-012420-001, Rev M [May 2018])

    See *Entrust nShield e-Security Integration Guide for Oracle WebLogic Server for UNIX, and Entrust nShield e-Security Cryptographic API Integration Guide*.

## Adaptable Application

CyberArk builds and delivers software drivers designed to connect with many of the most common certificate authorities (CAs), application servers, and enterprise monitoring systems for notification use cases. But if the CyberArk Driver Library does not include the driver you need, you can use a Venafi Adaptable Driver. In fact, because they are customizable, you can use an Adaptable driver in place of an existing driver to provide tighter integration between your business processes and Trust Protection Foundation.

Adaptable drivers provide a common set of variables required by the majority of applicable use cases that are supported by Trust Protection Foundation natively. Some Adaptable drivers also let you define additional text fields, yes/no (Boolean) fields, and a password credential field, which you can then use to elicit different behaviors or to pass additional data to the system or service to which you are integrating.

Adaptable drivers depend on a Microsoft PowerShell script hosted in your local environment to execute functions corresponding to standard certificate lifecycle stages or Trust Protection Foundation events.

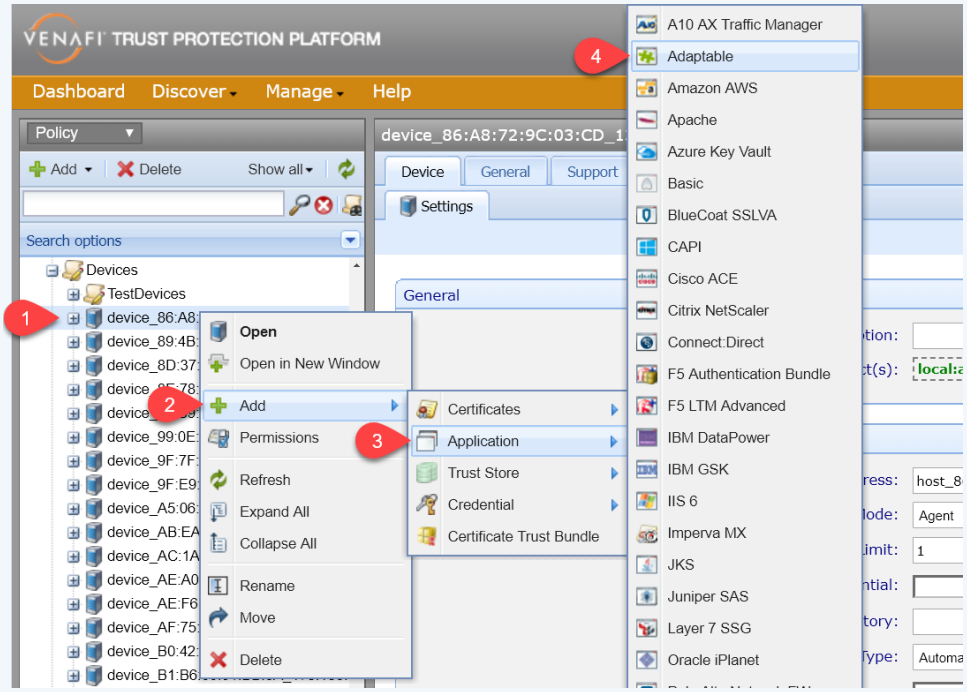
## Important Considerations

- Remote key generation is not performed in cases where a single certificate is associated with multiple applications through an Adaptable Application driver.
- A working knowledge of PowerShell scripting or equivalent experience is required to work effectively with Adaptable Driver solutions.

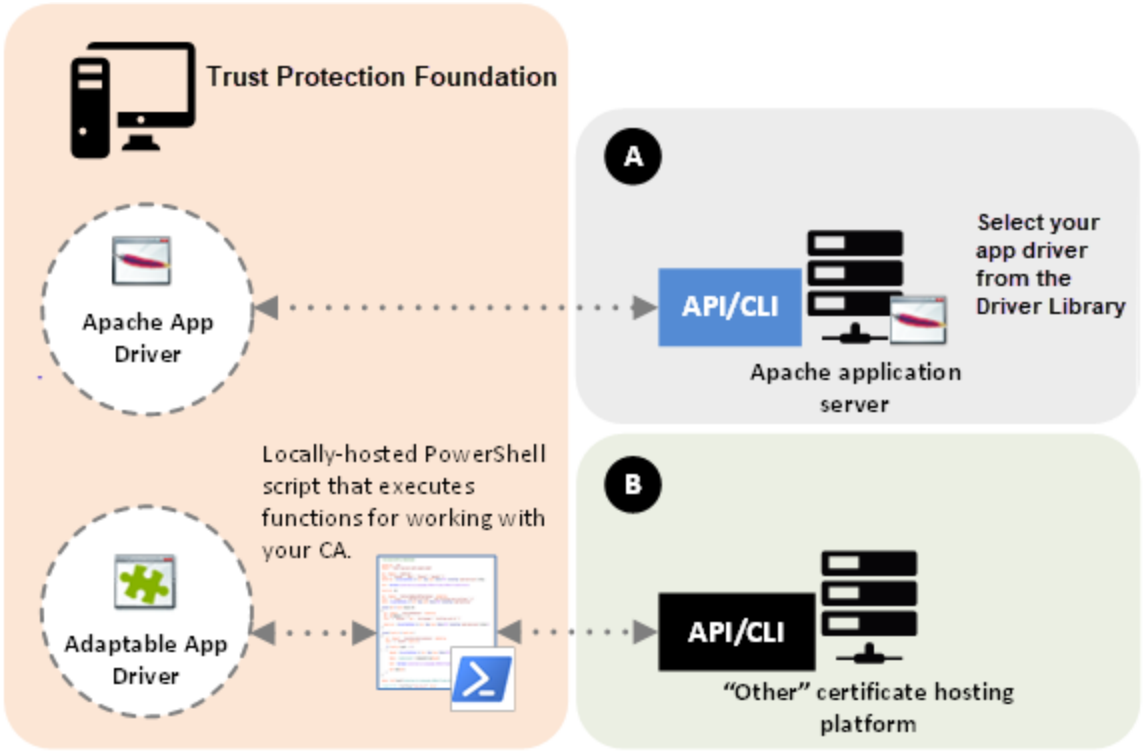
## Example

If you want to automate certificate provisioning, you can configure Trust Protection Foundation to install certificates for use by an application server . This is done using a CyberArk application object that matches your specific application.

**EXAMPLE** Suppose you want to automate certificate provisioning for your *MyAppliance* server. In the Policy Tree, you would first create and configure a device object for your MyAppliance server, and then you would create and configure a MyAppliance application object.




Finally, you would associate the application object with the relevant certificates used by your application server software.




## What's next?

Before implementing Adaptable Application drivers, carefully review prerequisites and learn more about the required PowerShell scripting.

For more information, see "Adaptable Application prerequisites" below and "About Venafi Adaptable Driver PowerShell scripts" on the next page.

## Adaptable Application prerequisites

Before you configure a Venafi Adaptable Driver application object in Trust Protection Foundation, review the following prerequisites:

- Review the Adaptable Flow design to understand where you want the custom logic to be applied.
- To work effectively with any CyberArk adaptable solution, you must have some working knowledge of PowerShell scripting, or you must have equivalent experience with a scripting language similar to PowerShell.
- Install Windows PowerShell 3.0 or later.

- Install .NET 4.8 or later.
- Make sure you have a username credential that is valid for the device you will connect to. This is the username and password SSH Manager for Machines will use to connect to the device. For information on working with credentials, see Creating user name or password credentials.
- Check out the sample script in the `\Venafi\Scripts\AdaptableSSHManagement\sample\` folder. This will help you create your script. Your scripts should be stored in the `\Venafi\Scripts\AdaptableSSHManagement\` folder. You will only be able to select items that appear in that specific folder, not any descendant folders.

## What's next?

After you've completed the Adaptable Application prerequisites, the next thing you need to do is to assign the PowerShell script to the policy. For more information, see "Assigning the PowerShell script to a policy" on page 283.

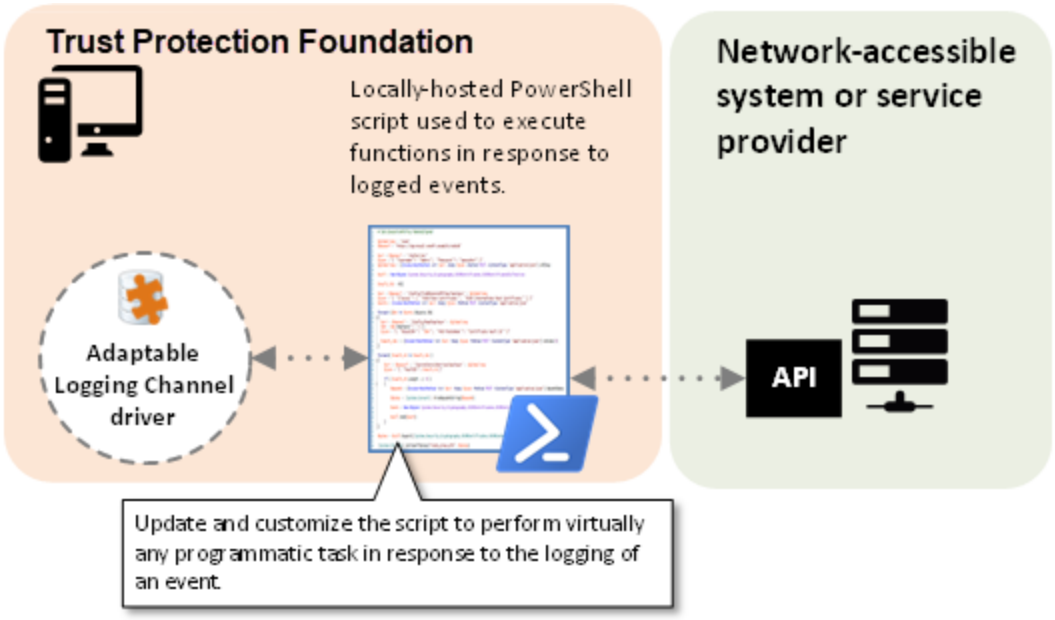
## About Venafi Adaptable Driver PowerShell scripts

Venafi Adaptable Drivers utilize PowerShell scripts—one for each driver—that contains functions called during various stages of the certificate lifecycle; or in the case of the Adaptable Log Channel, that can be used to perform virtually any programmatic task in response to the logging of a Trust Protection Foundation event.

For the Adaptable Application driver, the script can only be defined at the policy level—and is locked automatically—by selecting it from a drop-down list. After you've selected a script, it appears in a read-only field on the associated object in Trust Protection Foundation.

> **TIP** When creating new PowerShell scripts for use with adaptable drivers, keep in mind that the file name is used to identify your script from within the associated object (and within the policy for the Adaptable Application driver) in Trust Protection Foundation. Using logical names can help you and other administrators recognize the purpose and intent of each script.

Figure 4.2: PowerShell scripts include functions that work with Venafi Adaptable Drivers and the certificate hosting platform.




When using the Adaptable Application or Adaptable Log Channel drivers, you can define up to 12 text fields, two yes/no (Boolean) fields, and one password credential field in your PowerShell script. Fields are defined near the top of each PowerShell script (in the *Begin Field Definitions* section).

As part of configuring your fields, you can specify properties for each one:

- Should the field be enabled (visible to the user)?
- Can the field's value be applied using a policy? and
- Should the field require users to enter a value?

You set each property by simply typing a zero (0) to disable or a one (1) to enable it. So, for example:

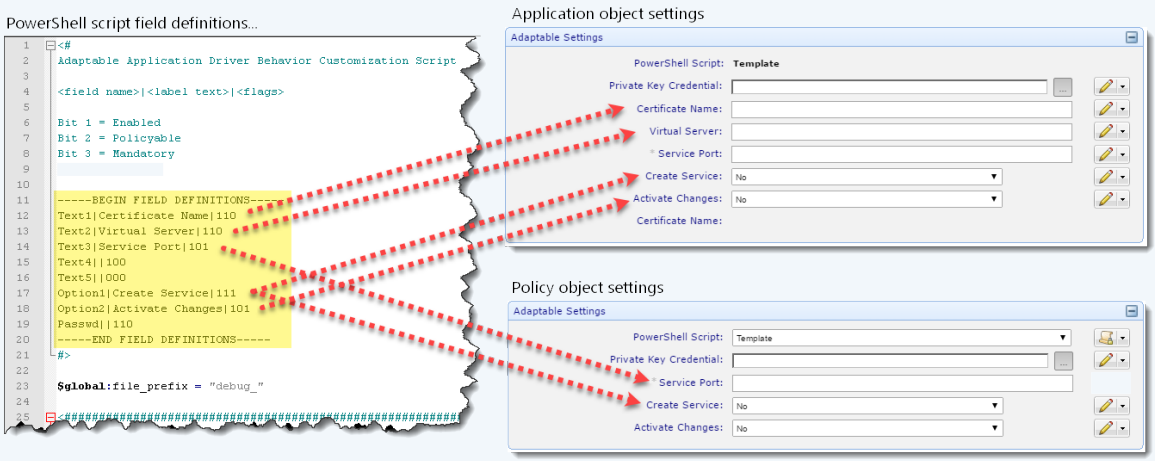
```
Text1|Certificate Name|110
```

In this case, a field would be presented to the user labeled "Certificate Name" (1=enabled) and would be policyable (1=enabled), but *not* mandatory (0=disabled).

**IMPORTANT** If there are only two bits, the first bit defines visibility and the second bit defines if the value is required.

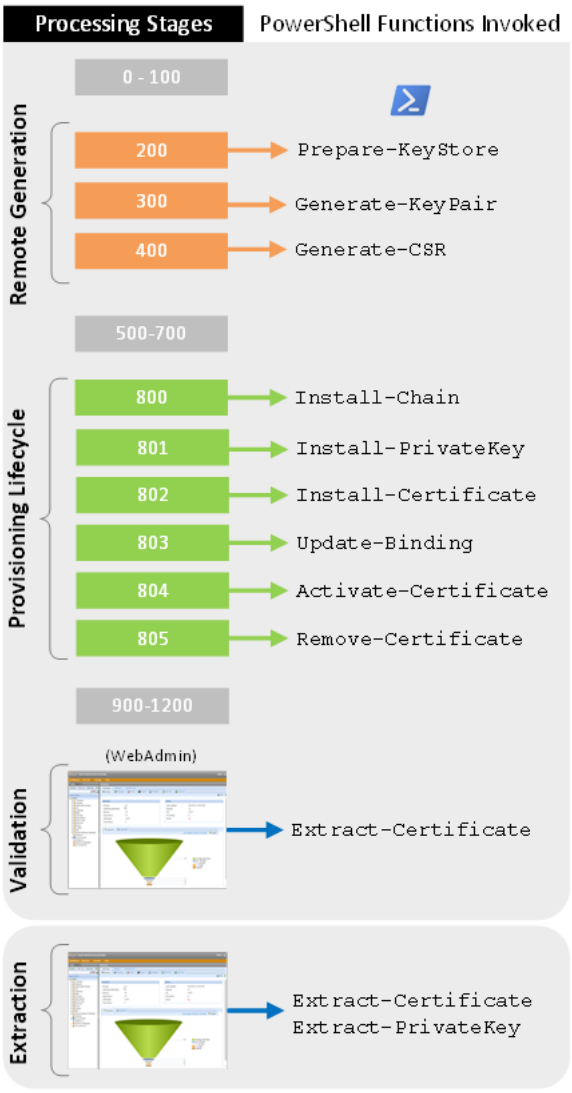
Once you save your script to the correct directory on the Trust Protection Foundation server, your new fields appear in the related object in Trust Protection Foundation (and for the Adaptable Application, in the related policy).

**EXAMPLE** Here's an example of field definitions defined in a PowerShell script used with the Adaptable Application driver and how they appear in Trust Protection Foundation based on property settings:




The input parameters and response format for each function is predefined. All functions receive a set of general parameters, whereas those parameters that are *specific* to the function are only passed to it. All of the standard provisioning functions must be present in your script, but the Remote Generation functions are optional and their absence is used to indicate a lack of support for "Generate Key/CSR on Application=Yes".

Functions are called at specific stages of the provisioning lifecycle. For more information, see "PowerShell script reference for Adaptable Application" on page 291.




## Why Windows PowerShell?

Trust Protection Foundation runs on the Windows Server operating system where PowerShell is already included. PowerShell is also an extremely flexible programming language because it can execute PowerShell *cmdlets* (both native and third party), .NET classes, and any binary that runs on Windows.

By using PowerShell, scripts that are to be used by the Adaptable Application driver can be developed and tested independently of Trust Protection Foundation, making integration simpler.

## Using the sample HP iLO and Dell iDRAC PowerShell scripts

When you install Trust Protection Foundation, a PowerShell reference sample for use with HP iLO and Dell iDRAC is included in the `\Venafi\Scripts\AdaptableApp\Samples` folder. You can use these reference samples to better understand how to configure your own scripts.

The reference samples are a *fully-functional implementation* that lets you use the Adaptable Application driver with HP iLO or Dell iDRAC. You do not need to know anything about the PowerShell functions to begin using the reference samples. However, you do need to set up your HP or Dell accounts correctly and modify a few variables in the script.

**NOTE** Both samples rely on management utilities provided by their respective vendors. While newer versions of iLO and iDRAC added REST API support, CyberArk opted to use the vendor's utilities with the sample scripts in order to provide the broadest coverage, including potential compatibility with older versions.

The scripts themselves explain their requirements in their respective comment headers and identify the required vendor software you need to install on the Trust Protection Foundation server.

**IMPORTANT** While these scripts might be compatible with older iLO and iDRAC firmware versions than stated above, firmware issues are the most common cause of script failures. Therefore, should you encounter any certificate provisioning issues, we strongly encourage you to update to the latest firmware versions.

```
//-----------------------------------------------------------------------
// This script relies upon the HPQLOCFG utility for remote administration
// HP iLO and was successfully tested using HPQLOCFG 5.2.0 and iLO 4 firmware
// version 2.50.
//
// https://support.hpe.com/hpsc/swd/public/detail?swItemId=MTX_8abe539b67bf46978e8f84acb8
//-----------------------------------------------------------------------
```

```
// -----------------------------------------------------------------------
// This script relies upon the RACADM utility for remote administration
// and was successfully tested using RACADM 8.3 and iDRAC 8 firmware
// version 2.41.40.40.
//
// http://www.dell.com/support/home/us/en/04/Drivers/DriversDetails?driverId=6RWMR
//-----------------------------------------------------------------------
```

## Assigning the PowerShell script to a policy

The PowerShell script must be assigned to a policy. This makes it easier to then share your Adaptable Application settings and script across many applications without having to specify the same settings many times, which is a key benefit of policies.

> **NOTE** The PowerShell script setting on the policy is always locked.

After you have selected your PowerShell script from the Adaptable Application's policy settings page, then the name of your script appears on the application object as a read-only field. If you need to select a different PowerShell script, do so from the policy. See "Selecting an alternate PowerShell script on a policy" on the next page.

> **BEST PRACTICE** You should assign your PowerShell script to policies that are as low down in the Policy tree as possible to allow you to make use of multiple Adaptable Application solutions and also to avoid inheritance issues with extra fields that are likely to occur if your PowerShell scripts are ever changed. If you set it at the top of the tree, then you are essentially committing to use only one Adaptable Application solution in your environment.
>
> For more information and best practices related to working with policies, see How policies work in the *Trust Protection Foundation Administration Guide*.

### To assign the PowerShell script to a policy

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. In the **Policy** tree, click the policy object where you want to specify the PowerShell script.
3. On the policy page, click the **Applications** tab, and then click **Adaptable**.
4. Under **Adaptable Settings**, click the **PowerShell Script** drop-down list and select the script you want to use.
5. (Optional) If you want to set the **General**, **Application Information**, and **Validation** settings at the policy level, do so now.

   For details about these settings, see "Configuring an Adaptable Application object" on page 285.
6. Click **Save**.

   > **TIP** When you click Save, the policy is locked automatically.

## What's Next?

Now that you have set the PowerShell script on the policy, you can create (or modify an existing) Adaptable Application application object.

For more information, see "Configuring an Adaptable Application object" on the facing page.

## Selecting an alternate PowerShell script on a policy

If you need to select a different PowerShell script for Adaptable Application applications, do so at the policy but it is very important to understand the impact of changing the script assignment. After selecting an alternate PowerShell script from the Adaptable Application policy settings page, any changes to the field definitions will then appear in Policy Tree. However, any Adaptable Application data that has been previously entered on child policies or Adaptable Application objects will remain unchanged and may not be applicable to the current PowerShell script.

> **BEST PRACTICE** You should assign your PowerShell script to policies that are as low down in the Policy tree as possible to allow you to make use of multiple Adaptable Application solutions and also to avoid inheritance issues with extra fields that are likely to occur if your PowerShell scripts are ever changed. If you set it at the top of the tree, then you are essentially committing to use only one Adaptable Application solution in your environment.
>
> For more information and best practices related to working with policies, see How policies work in the *Trust Protection Foundation Administration Guide*.

You can also verify that your new PowerShell script is being referenced by looking at the read-only PowerShell Script field on an Adaptable Application application object. The PowerShell Script setting on the policy is always locked.

### To use an alternate PowerShell script on a policy

1. From the Certificate Manager - Self-Hosted menu bar, click Policy Tree.
2. In the **Policy** tree, click the policy object where you want to assign an alternate PowerShell script.
3. On the policy page, click the **Applications** tab, and then click **Adaptable**.
4. Under **Adaptable Settings**, click the **PowerShell Script** drop-down list and select the script you want to use.

5. (Optional) If you want to modify the **General**, **Application Information**, and **Validation** settings at the policy level, do so now.

   For details about these settings, see "Configuring an Adaptable Application object" below.

6. Click **Save**.

### What's Next?

Now that you have selected an alternate PowerShell script on the policy, you can create (or modify an existing) Adaptable Application application object.

For more information, see "Configuring an Adaptable Application object" below.

## Configuring an Adaptable Application object

As with other CyberArk application drivers, you should review the "Getting started: automating certificate enrollment and provisioning" on page 15 section before setting up the Adaptable Application driver.

Creating and configuring an Adaptable Application object is similar to creating any other application object, except that you must specify the PowerShell script you want to use at the policy level. Before you create your Adaptable Application application object, see "Assigning the PowerShell script to a policy" on page 283.

**DID YOU KNOW?** When you add an installation to a certificate, you'll have the option of defining (and editing) this object during that process, which means that you don't have to log in to Policy Tree as the following procedure describes. And because the settings are the same, you can use this topic for information about each setting.

For more information, see Creating a certificate installation in Aperture in the *Certificate Management Guide*.

**BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

**NOTE** Before you attempt to create CA template, device, or application objects, you must enable the *create* permission under the folder where you want to create the new object. For more information, see Permissions Overview in the *Administration Guide*.

### To create and configure a new Adaptable Application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **Adaptable Application**.
3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.
4. (Optional) In the **Device Certificate** box, click ... to select and associate a certificate with the new application.

   > **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

   For detailed information on associating a certificate with an application, see Associating a certificate with an application object.
5. Under **General**, do the following:
   a. In the **Application Name** field, type a name for the new application.
   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.
   c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

      Default system notifications are sent to the contact identities. The default contact is the master administrator.

> **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
>
> Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

   The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

   You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

   For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Under **Application Information**, do the following:

   a. Click  next to **Application Credential** to browse for the credential object that you want to use to authenticate with the application.

   > **DID YOU KNOW?** Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential might be a user name or private key credential; some drivers–such as F5, which is not SSH-based–can only use the user name credential for authentication.

   > **NOTE** The user account you select must have *Read* and *Write* access to the Temporary, Private Key, and Certificate directories.
   >
   > If you need help with this step, see your system administrator.
   >
   > For more information, see Working with system credentials in the *Administration Guide*.

b. (Optional) (Conditional) If you need to select another credential, then from the **Secondary Credential** field, select a username, certificate, password, or CyberArk credential object.

> **TIP** Use this option to avoid having to hard code additional credentials in your script or having to utilize other solutions outside of Trust Protection Foundation.

c. (Optional) In the **Port** field, type the port that Trust Protection Foundation should use to communicate with the server where the application is installed.

Trust Protection Foundation uses the SSH protocol to communicate with the application server installed on Linux or Windows. The default SSH port assignment is port 22.

7. Under **Adaptable Settings**, do the following:

a. (Read Only) If the **PowerShell Script** setting is blank, then you have not yet specified a script on this application object's parent policy.

You can complete this procedure and then select the script in the policy later.

For more information, see "Assigning the PowerShell script to a policy" on page 283.

b. (Conditional)(Optional) If you did not select a private key credential on the parent policy already, or if you want to override the policy setting, click the **Private Key Credential** field and select one.

**Why do I need to select a private key credential?** The private key credential is the password used to encrypt the private key in the encrypted form passed to the function (both PEM and PKCS#12). If not supplied, we'll generate a password internally for that purpose because it may only be needed to install the private key (i.e. the private is decrypted by the consumer when you import it). You can also set this on the policy level of that object.

c. (Optional) If you want to enhance troubleshooting capabilities of your Adaptable ApplicationAdaptable Flow, select the **Enable Debug Logging** check box.

For information about how enabling this option works with the PowerShell script, see "About debug logging" on page 294 in the Adaptable Application Adaptable Flow PowerShell script reference.

d. In the **Pkcs#12 Encryption Algorithm** field, you can choose the encryption algorithm used when exporting PKCS#12 files. Stronger algorithms are usually less compatible, especially with older software.

e. Complete any remaining fields, as defined by your PowerShell script.

8. When you're finished, click **Save**.

## What's next?

After you've created an application object, here are other things you can do to manage the new application:

- On the application's **Settings** sub-tab:
  - Click  to push a certificate to its associated application.

    For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.
  - Click  **Reset** to stop processing the application and reset the status and stage.
  - Click  to reattempt installation of the certificate to its associated application, .
  - Click  **Validate Now** to validate the applications associated certificate.

    Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

    For more information, see "About certificate and application validation" on page 270.
- On the application object's **Validation** tab, you can configure validation settings for the application object.
- On an object's **General** tab:
  - Click the **Log** sub-tab to view any events that are triggered by the template object.
  - Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

## Protecting against unapproved changes to Adaptable Application scripts

CyberArk's Adaptable drivers–such as Adaptable CA, Adaptable Application, Adaptable Bulk Provisioning, and Adaptable Log Channel–rely on PowerShell scripts stored in the *\Venafi\Scripts\AdaptableDriverName* directory on Trust Protection Foundation servers. To ensure the integrity of these scripts, Trust Protection Foundation supports signed scripts.

Because other people might have access to the server that is running Trust Protection Foundation, they could modify your PowerShell scripts without your knowledge, either accidentally or intentionally. However, with signed scripts, any modifications made to the script will result in a failed signature validation, and the script will not be executed.

To protect against unapproved changes to your scripts, Trust Protection Foundation monitors PowerShell script files that are being used by existing Adaptable objects. If a new script is used or a PowerShell script is modified on the file system, Trust Protection Foundation displays a warning and you'll need to re-validate the script.

This security feature helps to prevent potentially harmful modifications to your scripts from being run.

**IMPORTANT** Because of this security feature, following an upgrade to Trust Protection Foundation, you must take specific steps on all existing Adaptable objects in order for them to be re-enabled. Refer to the documentation for each Adaptable driver for details.

When you encounter an error stating that the associated Adaptable Application PowerShell script has been modified, you'll need to open the policy where you assigned the script and re-save the policy settings. When you do, the driver is then re-enabled and you can continue using the script.

After you've verified and approved the changes made to the script, Trust Protection Foundation can also be configured to automatically retry all failed issues caused by script changes. This feature is available for the Adaptable CA and Adaptable Application drivers.

### To re-enable an Adaptable Application driver following a change to its associated PowerShell script

1. Open and review the associated scripts to verify that they contain only approved changes.
2. Open the Policy Tree and then open the policy where you assigned the associated PowerShell script.

   **DID YOU KNOW?** Unlike other Adaptable drivers, PowerShell scripts associated with Adaptable Application drivers are specified at the policy level rather than on the application object itself. For more information, see "Assigning the PowerShell script to a policy" on page 283.
3. On the policy, do the following:
   a. Click **Applications** > **Adaptable**.
   b. (Optional) Select **When script is updated, fix related provisioning and discovery errors** if you want Trust Protection Foundation to fix failed discovery and provisioning errors automatically.

> **DID YOU KNOW?** Enabling the *When script is updated...* feature instructs Trust Protection Foundation to retry all failed issues automatically once you have verified and approved the changes made to the script. How you approve scripts is done differently for each Adaptable driver.
>
> **Example:**
>
> Suppose one of your Adaptable CA scripts is modified. You see a warning repeated across ten certificate renewal requests. They are all using the same Adaptable CA template, which depends on the modified script file. The driver blocked processing when it detected script changes, so the renewal process failed. You open the script file to verify that the changes are safe and accurate. Then you open the associated Adaptable CA template, run another validation, set **When script is updated...** to **Yes**, and then save the template again. Because you enabled the option, all ten certificates are retried automatically and processing completes successfully. If you hadn't enabled this setting, you would need to go to each of the ten certificates and rerun certificate renewals manually.
>
> The time required for the automated fix to be completed depends on the number of issues. But Trust Protection Foundation fixes the issues as quickly as it can.

c. Click **Save** at the bottom of the page.

## PowerShell script reference for Adaptable Application

This section documents all available PowerShell functions for use with the Adaptable Application driver. PowerShell scripts are stored in the \Venafi\Scripts\AdaptableApp folder.

The input parameters and response format for each function is predefined. All functions receive a set of general parameters, whereas those parameters that are *specific* to the function are only passed to it. All of the standard provisioning functions must be present in your script, but the Remote Generation functions are optional and their absence is used to indicate a lack of support for "Generate Key/CSR on Application=Yes".

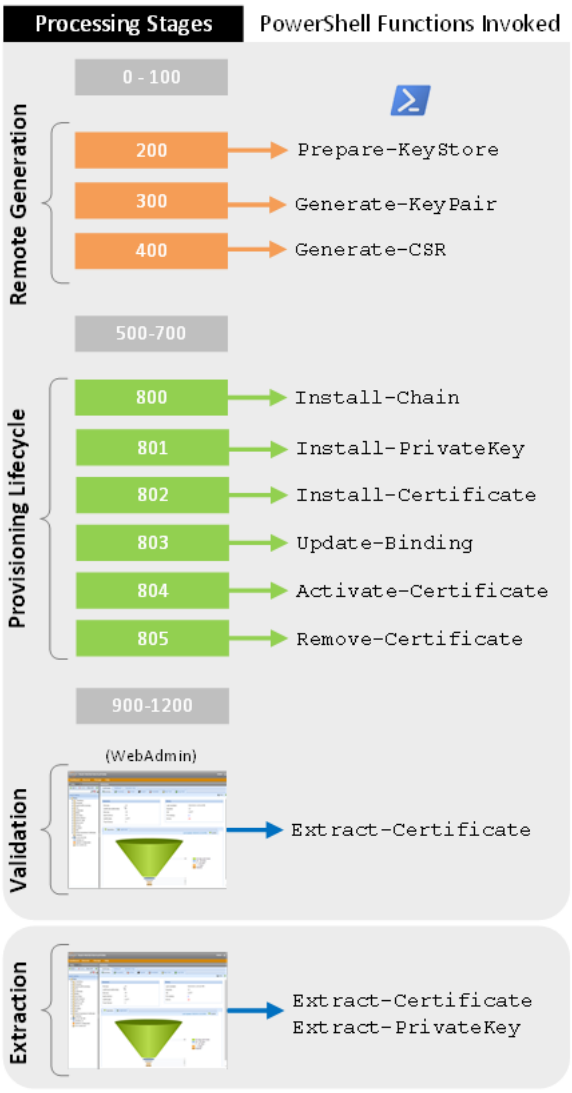
> **DID YOU KNOW?** To prevent vulnerabilities, the PowerShell scripts are stored on the Trust Protection Foundation server. While it might have been more convenient to allow downloading the script, storing the scripts on the Trust Protection Foundation server prevents potentially harmful scripts from affecting the server. Only privileged users on your Trust Protection Foundation server can access scripts.

You must ensure the same version of all your Adaptable Application scripts are on all servers in the cluster that have the WebConsole component installed. For this reason, it is wise to include a script version number in the file name, so you can easily check to see that the same version of the script is installed on all servers in the cluster.

> **NOTE** To work effectively with any CyberArk adaptable solution, you must have some working knowledge of PowerShell scripting, or you must have equivalent experience with a scripting language similar to PowerShell.

Data is passed to the functions using hash tables (key-value pairs). Using hash tables enables the addition of new functions in future releases. For more information, see "About hash tables for Adaptable Application" on page 296

The following diagram shows which PowerShell functions are invoked at which stage in the certificate lifecycle:




For information about processing stages, see "About certificate lifecycle management" on page 42.

**BEST PRACTICE** When customizing (or creating a new) PowerShell script, keep the following security best practices in mind:

- Avoid hard-coding credentials into your PowerShell scripts.
- Only include code in functions that relate to the task they are designated to perform.
- Scripts should not do anything that could alter the integrity or availability of the local Windows system (the system hosting Trust Protection Foundation).

## About debug logging

When a user has requested debug logging by checking **Enable Debug Logging** for Adaptable Application, the driver sets a global variable called $DEBUG_FILE whenever it executes a PowerShell function. So your PowerShell script should reference the value of the $DEBUG_FILE variable to decide whether or not to log information for troubleshooting purposes. The value the driver assigns to the $DEBUG_FILE variable is a recommended file path name on the Trust Protection Foundation server for use when logging events to a file. The file name is designed to be unique to the instance of the Adaptable component so as to avoid conflicts when multiple scripts are running at the same time and writing to the log file. If the recommended file name is used, the resulting log file appears in the `<Venafi Home>\Logs` directory by default (e.g. `C:\Program Files\Venafi\Logs`).

For information about where Enable Debug Logging is configured for Adaptable Application, see "Configuring an Adaptable Application object" on page 285, or if you're using Adaptable Application with Onboard Discovery, see Creating a new Onboard Discovery job.

| Variable Name | Data Type | Description |
|---|---|---|
| AssetName | String | The name used to uniquely identify the certificate that is provisioned to the device.<br><br>Value is initially automatically generated using the following naming convention:<br><br>■ *<Common Name>_<ValidTo as yyMMMdd>_<Last 4 of Serial>* for centrally generated CSRs<br><br>■ *<Common Name>_RGEN_<Current Date/Time as yyMMdd-HHmmss>* for remotely generated CSRs<br><br>AssetName can be overridden by several PowerShell functions if it is necessary for a particular device to use a different naming convention (e.g. to deal with string length or special character limitations). |
| AppObjectDN | String | Contains the Trust Protection Foundation distinguished name (DN) of the calling application object. |

| Variable Name | Data Type | Description |
|---|---|---|
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| HostAddress | String | Contains the hostname or IP address specified by the device object. |
| TcpPort | Integer | A value containing the TCP port specified by the application object. |
| UserName | String | The user name portion of the user name or private key credential assigned to the device or application object. Used for authenticating with the device. |
| UserPass | String | The password portion of the user name credential assigned to the device or application object. Used for authenticating with the device. |
| UserPrivKey | String | The privacy-enhanced electronic mail (PEM)-formatted RSA private key portion of the private key credential assigned to the device or application object. Used for authenticating with the device via SSH. |
| VarBool1 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarBool2 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarPass | String | Contains the value of the password field as defined by the header at the top of the PowerShell script. |

| Variable Name | Data Type | Description |
|---|---|---|
| VarText1 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText2 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText3 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText4 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText5 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |

## About hash tables for Adaptable Application

Data is passed to (and returned by) each function using hash table data structures.

A *general* hash table, which includes a common set of data, is passed to all of the functions. A *specific* hash table, which includes data that is applicable specifically to the function, is passed to functions that require additional data. All functions must return a single hash table that includes a *result* along with any other variables that the function is required to return.

Consider the following guidelines and expected behaviors:

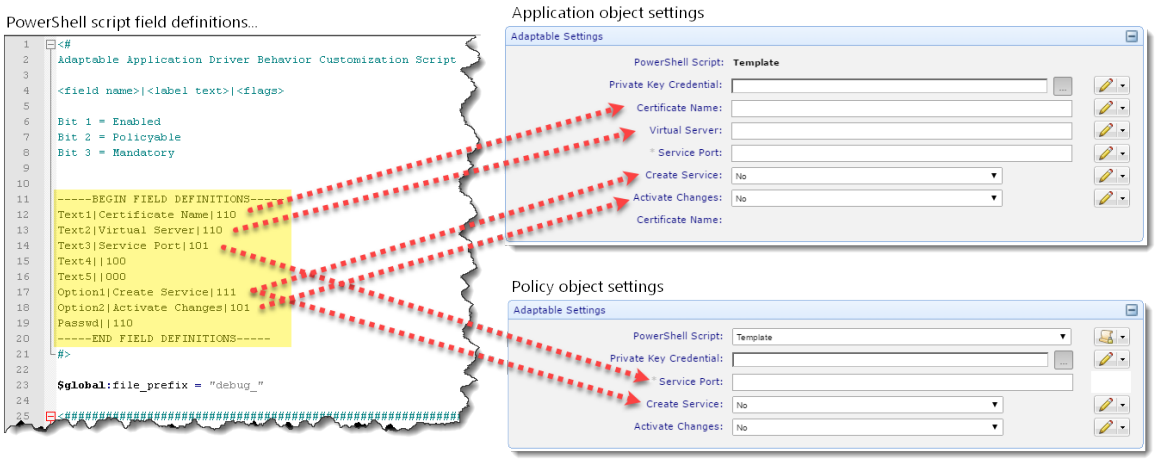
- Functions must not return errors; rather, they must throw exceptions in the same way that actual PowerShell errors do. The Adaptable CredentialAdaptable Application driver Adaptable SSH Key Discovery feature treats exceptions thrown by a PowerShell function as a fatal error and then halts processing.
- Thrown exceptions are handled as unexpected. If there is an error, we recommended you use `Result="Failure"` and pass the error description in the `Error=""` parameter.
- All functions (except where noted) must be present in the script, but they are optional from a logic standpoint.

- In the function references that follow, the following two functions are listed as not being required: `GetConfigPath`, and `CleanUP`. This means they are optional to implement. However, they should still be present in the script, and should return `Result="Success"` even with empty result data.
- The Generate-Keystore, Generate-KeyPair, and Generate-CSR functions should only be implemented if the target platform is to support remote generation of keys and CSR. The absence of any of these three functions will be interpreted by the driver as lack of support.
- The Generate-Keystore, Generate-KeyPair, Generate-CSR, Update-Binding, Activate-Certificate, Install-Chain, Install-Certificate, and Install-PrivateKey functions can return "ResumeLater" to indicate the stage should be re-processed at a later time.

  ResumeLater might be used when the endpoint to which the certificate is being installed takes longer to process the configuration or installation step then TLS protect allows for execution timeout

## Creating new data fields for Adaptable Application

The Adaptable Application configuration provides a common set of variables required by the majority of supported application servers and network appliances.. However, if you need to elicit different provisioning behaviors or pass unique or additional data to your application server, you can define up to five additional text fields, two yes/no (Boolean) fields, and one password credential field. You do this by simply modifying the **Begin Field Definitions** section of the Adaptable Application PowerShell script.




**NOTE** About macros: It's important to note that the Service Address field **does not accept macros**. You must use a custom field instead. Please be aware that certain macros are designed to function exclusively within workflows. Use the Macro Evaluator along with sample data to help you understand how macros will be processed for your use case.

Each field definition has three parts:

*Field Name*| *Field Label*| *Flags*

1. Field Name: Must be `Text1` through `Text5`, `Option1` and `Option2`, and `Passwd`. Do not modify these values.
2. Field Label: Type the name that you want to appear on the Adaptable Application tab of the policy, and on the Adaptable Application application object settings page.
3. Flags: Using the numeric values of `0` (to mean false) or `1` (to mean true) for each of the following options (in order):
   - Enabled: Use to enable or disable a field.
   - Policyable: Use to indicate whether a field can be set on a policy object.
   - Mandatory: Use to indicate whether the field must be completed, or is optional. When set to true, this option adds an asterisk (*) next to the field label to let the user know that it is a required field.

**EXAMPLE** Suppose you want to add a field for capturing the name of a certificate. Using a text (or PowerShell script) editor, you would open your PowerShell script file (which must be in the *drive*:\Program Files\Venafi\Scripts\AdaptableApp folder). You would then edit the field definitions in the header section to include a new field labeled Certificate Name and set values for the available flags:

```
----------BEGIN FIELD DEFINITIONS--------

Text1|Certificate Name|110
Text2|Text2|000
Text3|Text3|000
Text4|Text4|000
Text5|Text5|000
Option1|Create Service|111
Option2|Activate Changes|101
Passwd||110
```

When you are finished, you save your changes to the PowerShell script. When you open either Aperture or Policy Tree, your new fields appear on both the Adaptable Application tab on a policy object, and on the Adaptable Application application object.

**IMPORTANT** The header is required! The Powershell script will not work if the following header information is removed:

```
---- BEGIN FIELD DEFINITIONS ----
```

### To create (or edit) Adaptable Application data fields

1. Open your Adaptable Application  PowerShell file using a text (or PowerShell script) editor.

   Adaptable Application  PowerShell script files must be stored in the `\Venafi\Scripts\AdaptableApp\` folder.

2. Locate the **Begin Field Definitions** section and determine where you want to place your new field.

3. On the line where you want to define your new field (one of the Text or Option lines), type the new field label and then set the appropriate flags.

   For example:

   ```
   Text1|Certificate Name|110
   ```

4. When you are finished, save the file and close it.

Your new field should appear on both the Adaptable Application policy and application object pages in Policy Tree and Aperture.

## Discover-Certificates function

Use the *Discover-Certificates* function as part of your onboard discovery strategy to discover certificates found on network devices.

You can use all of the functions available for provisioning in your script; but the Discover-Certificates function is applicable to the Adaptable onboard discovery. You pass in the same general hash table (see below) with all of the connection and credentials

| Variable Name | Data Type | Description |
|---|---|---|
| AssetName | String | The name used to uniquely identify the certificate that is provisioned to the device.<br><br>Value is initially automatically generated using the following naming convention:<br><br>■ *<Common Name>_<ValidTo as yyMMMdd>_<Last 4 of Serial>* for centrally generated CSRs<br><br>■ *<Common Name>_RGEN_<Current Date/Time as yyMMdd-HHmmss>* for remotely generated CSRs<br><br>AssetName can be overridden by several PowerShell functions if it is necessary for a particular device to use a different naming convention (e.g. to deal with string length or special character limitations). |
| AppObjectDN | String | Contains the Trust Protection Foundation distinguished name (DN) of the calling application object. |
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| HostAddress | String | Contains the hostname or IP address specified by the device object. |
| TcpPort | Integer | A value containing the TCP port specified by the application object. |

| Variable Name | Data Type | Description |
|---|---|---|
| UserName | String | The user name portion of the user name or private key credential assigned to the device or application object. Used for authenticating with the device. |
| UserPass | String | The password portion of the user name credential assigned to the device or application object. Used for authenticating with the device. |
| UserPrivKey | String | The privacy-enhanced electronic mail (PEM)-formatted RSA private key portion of the private key credential assigned to the device or application object. Used for authenticating with the device via SSH. |
| VarBool1 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarBool2 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarPass | String | Contains the value of the password field as defined by the header at the top of the PowerShell script. |
| VarText1 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText2 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText3 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText4 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText5 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |

| Return | Data Type | Description |
|---|---|---|
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state. |
| Applications | Array of hashtables | An array of hashtables that include the following data for a certificate instance:<br>■ Name: the name of the application object that uniquely represents this instance under the device object<br>■ ApplicationClass: the schema class of the application object (optional, defaults to Adaptable App)<br>If an invalid class is specified, then a Basic application is created and an error event is logged for the onboard discovery job. If a name collision occurs with a Basic application, it is then converted to the specified application class (if valid).<br>For the Adaptable class and attribute names, see Adaptable Application driver in the Developer's Guide. For all other class and attribute names, see Object class reference in the Developer's Guide.<br>■ DriverName: the value of the Driver Name attribute (optional for application classes that are built into Trust Protection Foundation)<br>■ PEM: the raw X509 certificate in base 64 encoded text format<br>■ ValidationAddress: FQDN, hostname, or IP address where the server is presenting the certificate<br>**TIP** If you want to disable SSL/TLS validation, simply do *not* specify a ValidationAddress.<br>■ ValidationPort: TCP port on which the server is pre-sending the certificate (integer between 1-65535; defaults to 443)<br>■ Attributes: a hashtable that specifies zero or more Config attribute name-value pairs to set on the application object<br>If an attribute is not valid for the class, or the value is not a distinguished name (DN) for a DN attribute, then no value is assigned to the application and an error event is logged for the onboard discovery job. |

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Activate-Certificate function

Use the *Activate-Certificate* function to do whatever is necessary to commit the certificate that was provisioned to the target device and have the application server start using it. This function is called at processing stage 804.

| Variable Name | Data Type | Description |
|---|---|---|
| AssetName | String | The name used to uniquely identify the certificate that is provisioned to the device.<br><br>Value is initially automatically generated using the following naming convention:<br><br>■ *<Common Name>_<ValidTo as yyMMMdd>_<Last 4 of Serial>* for centrally generated CSRs<br><br>■ *<Common Name>_RGEN_<Current Date/Time as yyMMdd-HHmmss>* for remotely generated CSRs<br><br>AssetName can be overridden by several PowerShell functions if it is necessary for a particular device to use a different naming convention (e.g. to deal with string length or special character limitations). |
| AppObjectDN | String | Contains the Trust Protection Foundation distinguished name (DN) of the calling application object. |
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |

| Variable Name | Data Type | Description |
|---|---|---|
| HostAddress | String | Contains the hostname or IP address specified by the device object. |
| TcpPort | Integer | A value containing the TCP port specified by the application object. |
| UserName | String | The user name portion of the user name or private key credential assigned to the device or application object. Used for authenticating with the device. |
| UserPass | String | The password portion of the user name credential assigned to the device or application object. Used for authenticating with the device. |
| UserPrivKey | String | The privacy-enhanced electronic mail (PEM)-formatted RSA private key portion of the private key credential assigned to the device or application object. Used for authenticating with the device via SSH. |
| VarBool1 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarBool2 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarPass | String | Contains the value of the password field as defined by the header at the top of the PowerShell script. |
| VarText1 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText2 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |

| Variable Name | Data Type | Description |
|---|---|---|
| VarText3 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText4 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText5 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |

| Return | Data Type | Description |
|---|---|---|
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state.<br><br>Use "ResumeLater" to retry the current stage after 30 seconds. The wait time is increased using the formula $N^2$ *30 where N is the number of times the ResumeLater has been used in a row for this object in this stage. For example, the second time it will resume after 120 seconds and the 5th time it will resume after 750 seconds. |

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Extract-Certificate function

Use the *Extract-Certificate* function to retrieve the certificate from the target device so that it can be validated and extracted to update the certificate in CyberArk Trust Protection Foundation. This function is called after a certificate has been successfully provisioned and whenever a user requests the certificate and private key to be extracted from the device. This function is required because it provides an important method for Trust Protection Foundation to verify that the certificate is properly installed on the device. The function may return the raw X509 certificate or the serial number and SHA1 thumbprint for the purposes of Validation but it must return the raw X509 certificate to support Extraction.

| Variable Name | Data Type | Description |
|---|---|---|
| AssetName | String | The name used to uniquely identify the certificate that is provisioned to the device.<br><br>Value is initially automatically generated using the following naming convention:<br><br>■ *<Common Name>_<ValidTo as yyMMMdd>_<Last 4 of Serial>* for centrally generated CSRs<br><br>■ *<Common Name>_RGEN_<Current Date/Time as yyMMdd-HHmmss>* for remotely generated CSRs<br><br>AssetName can be overridden by several PowerShell functions if it is necessary for a particular device to use a different naming convention (e.g. to deal with string length or special character limitations). |
| AppObjectDN | String | Contains the Trust Protection Foundation distinguished name (DN) of the calling application object. |
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| HostAddress | String | Contains the hostname or IP address specified by the device object. |
| TcpPort | Integer | A value containing the TCP port specified by the application object. |

| Variable Name | Data Type | Description |
|---|---|---|
| UserName | String | The user name portion of the user name or private key credential assigned to the device or application object. Used for authenticating with the device. |
| UserPass | String | The password portion of the user name credential assigned to the device or application object. Used for authenticating with the device. |
| UserPrivKey | String | The privacy-enhanced electronic mail (PEM)-formatted RSA private key portion of the private key credential assigned to the device or application object. Used for authenticating with the device via SSH. |
| VarBool1 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarBool2 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarPass | String | Contains the value of the password field as defined by the header at the top of the PowerShell script. |
| VarText1 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText2 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText3 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText4 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText5 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |

| Return | Data Type | Description |
|---|---|---|
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state. |
| CertPem | String | Raw X509 certificate in Base64 PEM format. |
| Serial | String | Serial Number of the X509 certificate referenced by AssetName that is used for Validation when CertPem in not returned. |
| Thumbprint | String | SHA1 Thumbprint of the X509 certificate referenced by AssetName that is used for Validation when CertPem is not returned. |

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Extract-PrivateKey function

Use the *Extract-PrivateKey* function to retrieve the private key from the target device so that it can be extracted to update the certificate object in CyberArk Trust Protection Foundation. This function is called whenever a user requests the certificate and private key to be extracted from the device.

| Return | Data Type | Description |
|---|---|---|
| EncryptPass | String | Password that must be used if the private key the function returns is encrypted |

| Variable Name | Data Type | Description |
|---|---|---|
| AssetName | String | The name used to uniquely identify the certificate that is provisioned to the device.<br><br>Value is initially automatically generated using the following naming convention:<br><br>■ *<Common Name>_<ValidTo as yyMMMdd>_<Last 4 of Serial>* for centrally generated CSRs<br><br>■ *<Common Name>_RGEN_<Current Date/Time as yyMMdd-HHmmss>* for remotely generated CSRs<br><br>AssetName can be overridden by several PowerShell functions if it is necessary for a particular device to use a different naming convention (e.g. to deal with string length or special character limitations). |
| AppObjectDN | String | Contains the Trust Protection Foundation distinguished name (DN) of the calling application object. |
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| HostAddress | String | Contains the hostname or IP address specified by the device object. |
| TcpPort | Integer | A value containing the TCP port specified by the application object. |
| UserName | String | The user name portion of the user name or private key credential assigned to the device or application object. Used for authenticating with the device. |

| Variable Name | Data Type | Description |
|---|---|---|
| UserPass | String | The password portion of the user name credential assigned to the device or application object. Used for authenticating with the device. |
| UserPrivKey | String | The privacy-enhanced electronic mail (PEM)-formatted RSA private key portion of the private key credential assigned to the device or application object. Used for authenticating with the device via SSH. |
| VarBool1 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarBool2 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarPass | String | Contains the value of the password field as defined by the header at the top of the PowerShell script. |
| VarText1 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText2 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText3 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText4 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText5 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |

| Return | Data Type | Description |
|---|---|---|
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state. |
| PrivKeyPem | String | Private key referenced by AssetName in RSA Base64 PEM format, encrypted or not. If encrypted, the EncryptPass provided by the Specific hash table must be used. |

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Generate-Keypair function

Use the *Generate-Keypair* function to remotely generate a public-private key pair on the target device. This function is called at processing stage 300 (during remote generation). Remote generation is considered *unsupported* when this function is omitted or commented out.

| Variable Name | Data Type | Description |
|---|---|---|
| KeySize | Integer | Key bit strength of the key pair to be generated |
| EncryptPass | String | The password to use if encrypting the remotely generated private key. |

| Variable Name | Data Type | Description |
|---|---|---|
| AssetName | String | The name used to uniquely identify the certificate that is provisioned to the device.<br><br>Value is initially automatically generated using the following naming convention:<br><br>■ *<Common Name>_<ValidTo as yyMMMdd>_<Last 4 of Serial>* for centrally generated CSRs<br><br>■ *<Common Name>_RGEN_<Current Date/Time as yyMMdd-HHmmss>* for remotely generated CSRs<br><br>AssetName can be overridden by several PowerShell functions if it is necessary for a particular device to use a different naming convention (e.g. to deal with string length or special character limitations). |
| AppObjectDN | String | Contains the Trust Protection Foundation distinguished name (DN) of the calling application object. |
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| HostAddress | String | Contains the hostname or IP address specified by the device object. |
| TcpPort | Integer | A value containing the TCP port specified by the application object. |
| UserName | String | The user name portion of the user name or private key credential assigned to the device or application object. Used for authenticating with the device. |

| Variable Name | Data Type | Description |
|---|---|---|
| UserPass | String | The password portion of the user name credential assigned to the device or application object. Used for authenticating with the device. |
| UserPrivKey | String | The privacy-enhanced electronic mail (PEM)-formatted RSA private key portion of the private key credential assigned to the device or application object. Used for authenticating with the device via SSH. |
| VarBool1 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarBool2 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarPass | String | Contains the value of the password field as defined by the header at the top of the PowerShell script. |
| VarText1 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText2 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText3 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText4 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText5 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |

| Return | Data Type | Description |
|---|---|---|
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state.<br><br>Use "ResumeLater" to retry the current stage after 30 seconds. The wait time is increased using the formula N² *30 where N is the number of times the ResumeLater has been used in a row for this object in this stage. For example, the second time it will resume after 120 seconds and the 5th time it will resume after 750 seconds. |
| AssetName | String | (Optional) If returned, this value replaces the unique identifier by which the certificate is referenced for subsequent operations. |

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Generate-CSR function

Use the *Generate-CSR* function to generate a certificate signing request remotely on the target device. This function is called at processing stage 400 (during remote generation) and must return a PKCS#10-formatted CSR. For cases where key pair and CSR are generated on the target device by a single method, only *Generate-CSR* is applicable (not *Generate-Keypair*). Remote generation is considered *unsupported* when this function is omitted or commented out.

| Variable Name | Data Type | Description |
|---|---|---|
| KeySize | Integer | Key bit strength of the key pair to be generated. |
| EncryptPass | String | The password to use if encrypting the remotely generated private key. |
| SubjectDN | Hash Table | Keyed by Subject DN component (CN, OU, O, L, ST, C) where values are strings except for OU which is a string array. |
| SubjAltNames | Hash Table | Keyed by SAN type (DNS, Email, IP, URI, UPN) where values are string arrays of the individual Subject Alternative Names. |
| HashAlg | String | The signature algorithm that the user has requested to be used when generating the CSR remotely (e.g. sha1 or sha256). |

| Variable Name | Data Type | Description |
|---|---|---|
| AssetName | String | The name used to uniquely identify the certificate that is provisioned to the device.<br><br>Value is initially automatically generated using the following naming convention:<br><br>■ *<Common Name>_<ValidTo as yyMMMdd>_<Last 4 of Serial>* for centrally generated CSRs<br><br>■ *<Common Name>_RGEN_<Current Date/Time as yyMMdd-HHmmss>* for remotely generated CSRs<br><br>AssetName can be overridden by several PowerShell functions if it is necessary for a particular device to use a different naming convention (e.g. to deal with string length or special character limitations). |
| AppObjectDN | String | Contains the Trust Protection Foundation distinguished name (DN) of the calling application object. |
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| HostAddress | String | Contains the hostname or IP address specified by the device object. |
| TcpPort | Integer | A value containing the TCP port specified by the application object. |
| UserName | String | The user name portion of the user name or private key credential assigned to the device or application object. Used for authenticating with the device. |

| Variable Name | Data Type | Description |
|---|---|---|
| UserPass | String | The password portion of the user name credential assigned to the device or application object. Used for authenticating with the device. |
| UserPrivKey | String | The privacy-enhanced electronic mail (PEM)-formatted RSA private key portion of the private key credential assigned to the device or application object. Used for authenticating with the device via SSH. |
| VarBool1 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarBool2 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarPass | String | Contains the value of the password field as defined by the header at the top of the PowerShell script. |
| VarText1 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText2 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText3 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText4 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText5 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |

| Return | Data Type | Description |
|---|---|---|
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state.<br><br>Use "ResumeLater" to retry the current stage after 30 seconds. The wait time is increased using the formula $N^2$ *30 where N is the number of times the ResumeLater has been used in a row for this object in this stage. For example, the second time it will resume after 120 seconds and the 5th time it will resume after 750 seconds. |
| AssetName | String | (Optional) If returned, this value replaces the unique identifier by which the certificate is referenced for subsequent operations. |
| Pkcs10 | String | Base64-encoded string representation of the CSR in the standard PKCS#10 format. |

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Install-Chain function

Use the Install-Chain function to provision each of the certificates in the certificate authority trust chain to the target device. This function is called at processing stage 800.

| Variable Name | Data Type | Description |
|---|---|---|
| ChainPem | String | All of the CA certificates in the trust chain for the end-entity certificate in Base64 PEM format concatenated together. |
| ChainPkcs7 | Byte Array | All of the CA certificates in the trust chain for the end-entity certificate in PKCS#7 format. |

| Variable Name | Data Type | Description |
|---|---|---|
| AssetName | String | The name used to uniquely identify the certificate that is provisioned to the device.<br><br>Value is initially automatically generated using the following naming convention:<br><br>■ *<Common Name>_<ValidTo as yyMMMdd>_<Last 4 of Serial>* for centrally generated CSRs<br><br>■ *<Common Name>_RGEN_<Current Date/Time as yyMMdd-HHmmss>* for remotely generated CSRs<br><br>AssetName can be overridden by several PowerShell functions if it is necessary for a particular device to use a different naming convention (e.g. to deal with string length or special character limitations). |
| AppObjectDN | String | Contains the Trust Protection Foundation distinguished name (DN) of the calling application object. |
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| HostAddress | String | Contains the hostname or IP address specified by the device object. |
| TcpPort | Integer | A value containing the TCP port specified by the application object. |
| UserName | String | The user name portion of the user name or private key credential assigned to the device or application object. Used for authenticating with the device. |

| Variable Name | Data Type | Description |
|---|---|---|
| UserPass | String | The password portion of the user name credential assigned to the device or application object. Used for authenticating with the device. |
| UserPrivKey | String | The privacy-enhanced electronic mail (PEM)-formatted RSA private key portion of the private key credential assigned to the device or application object. Used for authenticating with the device via SSH. |
| VarBool1 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarBool2 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarPass | String | Contains the value of the password field as defined by the header at the top of the PowerShell script. |
| VarText1 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText2 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText3 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText4 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText5 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |

| Return | Data Type | Description |
|---|---|---|
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state.<br><br>Use "ResumeLater" to retry the current stage after 30 seconds. The wait time is increased using the formula N² *30 where N is the number of times the ResumeLater has been used in a row for this object in this stage. For example, the second time it will resume after 120 seconds and the 5th time it will resume after 750 seconds. |

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Install-PrivateKey function

Use the *Install-PrivateKey* function to provision the private key associated with the certificate object to the target device. This function is called at processing stage 801 and must be implemented if *Install-Certificate* is not used.

| Variable Name | Data Type | Description |
|---|---|---|
| PrivKeyPem | String | The non-encrypted RSA private key being provisioned in Base64 PEM format. |
| PrivKeyPemEncrypted | String | The RSA private key being provisioned in Base64 PEM format and encrypted using the EncryptPass password. |
| EncryptPass | String | The password used to encrypt the PrivKeyPemEncrypted private key. |

| Variable Name | Data Type | Description |
|---|---|---|
| AssetName | String | The name used to uniquely identify the certificate that is provisioned to the device.<br><br>Value is initially automatically generated using the following naming convention:<br><br>■ *<Common Name>_<ValidTo as yyMMMdd>_<Last 4 of Serial>* for centrally generated CSRs<br><br>■ *<Common Name>_RGEN_<Current Date/Time as yyMMdd-HHmmss>* for remotely generated CSRs<br><br>AssetName can be overridden by several PowerShell functions if it is necessary for a particular device to use a different naming convention (e.g. to deal with string length or special character limitations). |
| AppObjectDN | String | Contains the Trust Protection Foundation distinguished name (DN) of the calling application object. |
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| HostAddress | String | Contains the hostname or IP address specified by the device object. |
| TcpPort | Integer | A value containing the TCP port specified by the application object. |
| UserName | String | The user name portion of the user name or private key credential assigned to the device or application object. Used for authenticating with the device. |

| Variable Name | Data Type | Description |
|---|---|---|
| UserPass | String | The password portion of the user name credential assigned to the device or application object. Used for authenticating with the device. |
| UserPrivKey | String | The privacy-enhanced electronic mail (PEM)-formatted RSA private key portion of the private key credential assigned to the device or application object. Used for authenticating with the device via SSH. |
| VarBool1 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarBool2 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarPass | String | Contains the value of the password field as defined by the header at the top of the PowerShell script. |
| VarText1 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText2 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText3 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText4 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText5 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |

| Return | Data Type | Description |
|---|---|---|
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state.<br><br>Use "ResumeLater" to retry the current stage after 30 seconds. The wait time is increased using the formula N² *30 where N is the number of times the ResumeLater has been used in a row for this object in this stage. For example, the second time it will resume after 120 seconds and the 5th time it will resume after 750 seconds. |
| AssetName | String | (Optional) If returned, this value replaces the unique identifier by which the certificate is referenced for subsequent operations. |

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Install-Certificate function

Use the *Install-Certificate* function to provision the end-entity certificate to the target device. This function is called at processing stage 802 and must be implemented if *Install-PrivateKey* is not used. For cases where the target device requires the certificate and private key to be installed at the same time, only *Install-Certificate* is applicable (not *Install-PrivateKey*) because it is passed the end-entity certificate, its corresponding private key, and its chain certificates.

| Variable Name | Data Type | Description |
|---|---|---|
| CertPem | String | End-entity X509 certificate to be provisioned in Base64 PEM format. |
| PrivKeyPem | String | The non-encrypted RSA private key being provisioned in Base64 PEM format. |
| PrivKeyPemEncrypted | String | The RSA private key being provisioned in Base64 PEM format and encrypted using the EncryptPass password. |
| ChainPem | String | All of the CA certificates in the trust chain for the end-entity certificate in Base64 PEM format concatenated together |

| Variable Name | Data Type | Description |
|---|---|---|
| ChainPkcs7 | Byte Array | All of the CA certificates in the trust chain for the end-entity certificate in PKCS#7 format. |
| Pkcs12 | Byte Array | PKCS#12 keystore that includes certificate, private key, and chain |
| EncryptPass | String | Password used to encrypt the PrivKeyPemEncrypted private key and Pkcs12 keystore. |

| Variable Name | Data Type | Description |
|---|---|---|
| AssetName | String | The name used to uniquely identify the certificate that is provisioned to the device.<br><br>Value is initially automatically generated using the following naming convention:<br><br>■ *<Common Name>_<ValidTo as yyMMMdd>_<Last 4 of Serial>* for centrally generated CSRs<br><br>■ *<Common Name>_RGEN_<Current Date/Time as yyMMdd-HHmmss>* for remotely generated CSRs<br><br>AssetName can be overridden by several PowerShell functions if it is necessary for a particular device to use a different naming convention (e.g. to deal with string length or special character limitations). |
| AppObjectDN | String | Contains the Trust Protection Foundation distinguished name (DN) of the calling application object. |
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |

| Variable Name | Data Type | Description |
|---|---|---|
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| HostAddress | String | Contains the hostname or IP address specified by the device object. |
| TcpPort | Integer | A value containing the TCP port specified by the application object. |
| UserName | String | The user name portion of the user name or private key credential assigned to the device or application object. Used for authenticating with the device. |
| UserPass | String | The password portion of the user name credential assigned to the device or application object. Used for authenticating with the device. |
| UserPrivKey | String | The privacy-enhanced electronic mail (PEM)-formatted RSA private key portion of the private key credential assigned to the device or application object. Used for authenticating with the device via SSH. |
| VarBool1 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarBool2 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarPass | String | Contains the value of the password field as defined by the header at the top of the PowerShell script. |
| VarText1 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText2 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |

| Variable Name | Data Type | Description |
|---|---|---|
| VarText3 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText4 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText5 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |

| Return | Data Type | Description |
|---|---|---|
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state.<br><br>Use "ResumeLater" to retry the current stage after 30 seconds. The wait time is increased using the formula $N^2$ *30 where N is the number of times the ResumeLater has been used in a row for this object in this stage. For example, the second time it will resume after 120 seconds and the 5th time it will resume after 750 seconds. |
| AssetName | String | (Optional) If returned, this value replaces the unique identifier by which the certificate is referenced for subsequent operations. |

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Prepare-Keystore function

Use the *Prepare-Keystore* function to remotely create and/or verify the readiness of the keystore used by the target device, if applicable. This function is called at processing stage 200 (during remote generation). Remote generation is considered *unsupported* when this function is omitted or commented out.

| Variable Name | Data Type | Description |
|---|---|---|
| AssetName | String | The name used to uniquely identify the certificate that is provisioned to the device.<br><br>Value is initially automatically generated using the following naming convention:<br><br>■ *<Common Name>_<ValidTo as yyMMMdd>_<Last 4 of Serial>* for centrally generated CSRs<br><br>■ *<Common Name>_RGEN_<Current Date/Time as yyMMdd-HHmmss>* for remotely generated CSRs<br><br>AssetName can be overridden by several PowerShell functions if it is necessary for a particular device to use a different naming convention (e.g. to deal with string length or special character limitations). |
| AppObjectDN | String | Contains the Trust Protection Foundation distinguished name (DN) of the calling application object. |
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| HostAddress | String | Contains the hostname or IP address specified by the device object. |
| TcpPort | Integer | A value containing the TCP port specified by the application object. |

| Variable Name | Data Type | Description |
|---|---|---|
| UserName | String | The user name portion of the user name or private key credential assigned to the device or application object. Used for authenticating with the device. |
| UserPass | String | The password portion of the user name credential assigned to the device or application object. Used for authenticating with the device. |
| UserPrivKey | String | The privacy-enhanced electronic mail (PEM)-formatted RSA private key portion of the private key credential assigned to the device or application object. Used for authenticating with the device via SSH. |
| VarBool1 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarBool2 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarPass | String | Contains the value of the password field as defined by the header at the top of the PowerShell script. |
| VarText1 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText2 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText3 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText4 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText5 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |

| Return | Data Type | Description |
|---|---|---|
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state.<br><br>Use "ResumeLater" to retry the current stage after 30 seconds. The wait time is increased using the formula N² *30 where N is the number of times the ResumeLater has been used in a row for this object in this stage. For example, the second time it will resume after 120 seconds and the 5th time it will resume after 750 seconds. |

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Remove-Certificate function

Use the *Remove-Certificate* function to enable clean up the target device by removing past versions of the certificate, a feature known as "Generational Management". This function is called at processing stage 805 when CyberArk Trust Protection Foundation has provisioned the certificate at least three times.

| Return | Data Type | Description |
|---|---|---|
| AssetNameOld | String | The name of the past version of the certificate that is to be deleted. |

| Variable Name | Data Type | Description |
|---|---|---|
| AssetName | String | The name used to uniquely identify the certificate that is provisioned to the device.<br><br>Value is initially automatically generated using the following naming convention:<br><br>■ *<Common Name>_<ValidTo as yyMMMdd>_<Last 4 of Serial>* for centrally generated CSRs<br><br>■ *<Common Name>_RGEN_<Current Date/Time as yyMMdd-HHmmss>* for remotely generated CSRs<br><br>AssetName can be overridden by several PowerShell functions if it is necessary for a particular device to use a different naming convention (e.g. to deal with string length or special character limitations). |
| AppObjectDN | String | Contains the Trust Protection Foundation distinguished name (DN) of the calling application object. |
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| HostAddress | String | Contains the hostname or IP address specified by the device object. |
| TcpPort | Integer | A value containing the TCP port specified by the application object. |
| UserName | String | The user name portion of the user name or private key credential assigned to the device or application object. Used for authenticating with the device. |

| Variable Name | Data Type | Description |
|---|---|---|
| UserPass | String | The password portion of the user name credential assigned to the device or application object. Used for authenticating with the device. |
| UserPrivKey | String | The privacy-enhanced electronic mail (PEM)-formatted RSA private key portion of the private key credential assigned to the device or application object. Used for authenticating with the device via SSH. |
| VarBool1 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarBool2 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarPass | String | Contains the value of the password field as defined by the header at the top of the PowerShell script. |
| VarText1 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText2 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText3 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText4 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText5 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |

| Return | Data Type | Description |
|---|---|---|
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state. |

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Update-Binding function

Use the Update-Binding function to associate the provisioned certificate and private key to the consuming application on the target device. This function is called at processing stage 803.

| Variable Name | Data Type | Description |
|---|---|---|
| AssetName | String | The name used to uniquely identify the certificate that is provisioned to the device.<br><br>Value is initially automatically generated using the following naming convention:<br><br>■ *<Common Name>_<ValidTo as yyMMMdd>_<Last 4 of Serial>* for centrally generated CSRs<br><br>■ *<Common Name>_RGEN_<Current Date/Time as yyMMdd-HHmmss>* for remotely generated CSRs<br><br>AssetName can be overridden by several PowerShell functions if it is necessary for a particular device to use a different naming convention (e.g. to deal with string length or special character limitations). |
| AppObjectDN | String | Contains the Trust Protection Foundation distinguished name (DN) of the calling application object. |
| AuxPass | String | The password portion of the Secondary Credential when a user name or a password credential is assigned, or the PKCS#12 password when a certificate credential is assigned |
| AuxPfxData | Byte Array | A PKCS#12 byte array that contains a client certificate and private key when a certificate credential is assigned as the Secondary Credential |
| AuxUser | String | The user name portion of the Secondary Credential when a user name credential is assigned |
| HostAddress | String | Contains the hostname or IP address specified by the device object. |

| Variable Name | Data Type | Description |
|---|---|---|
| TcpPort | Integer | A value containing the TCP port specified by the application object. |
| UserName | String | The user name portion of the user name or private key credential assigned to the device or application object. Used for authenticating with the device. |
| UserPass | String | The password portion of the user name credential assigned to the device or application object. Used for authenticating with the device. |
| UserPrivKey | String | The privacy-enhanced electronic mail (PEM)-formatted RSA private key portion of the private key credential assigned to the device or application object. Used for authenticating with the device via SSH. |
| VarBool1 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarBool2 | Boolean | The value of the Yes/No (true/false) user-defined field as defined by the header at the top of the PowerShell script. |
| VarPass | String | Contains the value of the password field as defined by the header at the top of the PowerShell script. |
| VarText1 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText2 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText3 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText4 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |
| VarText5 | String | The text contained in the user-defined field as defined by the header at the top of the PowerShell script. |

| Return | Data Type | Description |
|---|---|---|
| Result | String | Shows "Success" or "NotUsed" to indicate the non-error completion state.<br><br>Use "ResumeLater" to retry the current stage after 30 seconds. The wait time is increased using the formula N² *30 where N is the number of times the ResumeLater has been used in a row for this object in this stage. For example, the second time it will resume after 120 seconds and the 5th time it will resume after 750 seconds. |

For information about processing stages, see "About certificate lifecycle management" on page 42.

## Amazon Web Services (AWS)–Overview

The Amazon Web Services (AWS) secure cloud services platform provides computing power, database storage, content delivery and other functionality that helps businesses scale and grow. Several AWS features utilize certificates to secure communications between AWS and clients.

CyberArk Trust Protection Foundation™ supports traditional provisioning of certificates and keys to three of the more popular AWS features:

- Elastic Load Balancing (both Application [ALB] and Classic [ELB])
- CloudFront
- IAM

Alternatively, Trust Protection Foundation supports the use of certificates issued by the Amazon Certificate Manager for ALB, ELB, CloudFront, and IAM. To learn more, visit the following URLs:

- https://aws.amazon.com/elasticloadbalancing/
- https://aws.amazon.com/cloudfront/
- https://aws.amazon.com/iam/

This chapter contains the following topics:

# Amazon AWS certificate and key algorithm considerations

Trust Protection Foundation can issue certificates using a wide range of key algorithms and key sizes. However, AWS places service-specific limits on which certificates can be accepted, stored, and used by its cloud services. These AWS platform restrictions may affect provisioning outcomes regardless of which algorithms are available in Trust Protection Foundation. This topic summarizes key considerations and highlights common limitations encountered when provisioning certificates to AWS services.

## Key generation models and algorithm impact

Trust Protection Foundation supports both central key generation (keys generated and stored in Trust Protection Foundation) and remote key generation (keys created by Amazon Certificate Manager). AWS algorithm rules apply regardless of generation method:

- Central generation: Trust Protection Foundation imports the certificate to ACM or IAM depending on the AWS service requirements.
- Remote generation: Amazon CA generates and stores the private key in ACM; Trust Protection Foundation references the certificate without uploading key material.

AWS service restrictions may block or limit certain combinations of key type, key size, and destination (ACM or IAM). Trust Protection Foundation detects these issues during provisioning and displays user-friendly validation messages where possible.

## Service-specific certificate and algorithm behaviors

AWS imposes different certificate requirements across its services. The following examples highlight commonly encountered limitations. For complete and current details, refer to AWS documentation.

#### Elastic Load Balancer (ELB – Classic Load Balancer)

- EC (ECDSA) and RSA-4096 certificates must be uploaded to IAM.
- Other RSA key sizes (such as RSA-2048 and RSA-3072) may be provisioned through ACM.
- Attempting to use EC or RSA-4096 certificates via ACM for ELB may cause provisioning failures.

#### CloudFront

CloudFront enforces stricter certificate requirements for viewer TLS connections:

- RSA-2048 is supported natively when the certificate is stored in ACM.
- Larger RSA keys may require generating the certificate externally and importing it into ACM.
- For ECDSA, CloudFront supports ECDSA P-256 (prime256v1) for viewer HTTPS. Other EC curves may not be accepted, even if ACM allows the certificate to be imported.

For details, see the CloudFront requirements for public key sizes and algorithms.

#### Application Load Balancer (ALB)

- ALB supports certificates provisioned through ACM, including RSA 2048/3072/4096 and ECDSA keys.
- ALB does not impose the ACM/IAM restrictions seen in ELB, making ALB the most flexible target for EC keys provisioned via ACM.

### Where to find AWS's authoritative key and certificate requirements

For complete AWS guidance on supported key types and sizes, see the AWS Key Management documentation:

- AWS Key Specification Reference

For CloudFront certificate and algorithm requirements, refer to:

- CloudFront HTTPS and public key requirements

## AWS certificate lifecycle

The following table outlines certificate lifecycle stages for certificates and private keys installed in the Amazon cloud and identifies the management level associated with each stage.

> Lifecycle stages 800 through 1200 are handled by application drivers. However, not all stages are defined for every driver. For more information about lifecycle stages, see "About certificate lifecycle management" on page 42 in the *CyberArk Trust Protection Foundation Certificate Management Guide*.

For the AWS driver, stages 800-802 are defined in the following table.

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 800 | InstallCertificate | Installs the certificate, private key, and the root and intermediate certificate chain. Applicable for non-Amazon Certificate Manager certificates. | | ✔ |
| 801 | BindCertificate | Assigns the certificate to an AWS Elastic Load Balancer listener, or an AWS CloudFront distribution. | | ✔ |
| 802 | GenerationalCleanup | Removes the certificate before the previous version of the certificate that was provisioned. Applicable for non-Amazon Certificate Manager certificates. | | ✔ |

## AWS permission requirements

API access to Amazon Web Services (AWS) is granted using two key credentials: an Access Key ID and a Secret Access Key. These credentials are created for a user by an AWS Identity and Access Management (IAM) administrator.

**BEST PRACTICE** Because these credentials are granted by an AWS IAM administrator, you should create a dedicated user account in Trust Protection Foundation for this purpose.

For more information, visit the following URL:

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html

The operations performed by Amazon Certificate Manager and Amazon Web Services drivers depend on a set of permissions found in your AWS policy. The following AWS policy represents the least privilege access required to support the full feature set of the CyberArkAWS driver.

```
{

        "Version": "2012-10-17",
        "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "iam:DeleteServerCertificate",
                "iam:UploadServerCertificate",
                "iam:ListServerCertificates",
                "iam:GetServerCertificate",
                "acm:RequestCertificate",
                "acm:GetCertificate",
                "acm:DeleteCertificate",
                "acm:ImportCertificate",
                "acm:ListCertificates",
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:DescribeListeners",
                "elasticloadbalancing:DescribeTargetGroups",
                "elasticloadbalancing:DescribeListenerCertificates",
                "elasticloadbalancing:CreateListener",
                "elasticloadbalancing:CreateLoadBalancerListeners",
                "elasticloadbalancing:ModifyListener",
                "elasticloadbalancing:SetLoadBalancerListenerSSLCertificate",
                "cloudfront:GetDistribution",
                "cloudfront:GetDistributionConfig",
                "cloudfront:UpdateDistribution",
                "cloudfront:ListDistributions",
                "ec2:DescribeInstances"
            ],

            "Resource": "*"
        }
    ]
}
```

You can remove specific features that you don't plan to use by simply omitting lines from the policy. For example, if you know that you will not be using ACM, delete any actions that start with "*acm:*".

For more information about ELB permissions, visit the following Amazon URL:

http://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_examples.html#d0e40339

## Creating an Amazon Web Services (AWS) application object

To install a certificate into the Amazon Web Services (AWS) ACM or IAM certificate store–and/or to bind a certificate to an Application Load Balancer (ALB), Elastic Load Balancer (ELB) or CloudFront distribution–you must apply the required configuration settings described in this topic.

**BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

As with other application drivers, make sure you first create the device object, which is the parent of application objects in Trust Protection Foundation. See "Creating a device object in the Policy Tree" on page 239 and "Device object settings (agent, agentless, and adaptable provisioning)" on page 240.

**DID YOU KNOW?** When you add an installation to a certificate, you'll have the option of defining (and editing) this object during that process, which means that you don't have to log in to Policy Tree as the following procedure describes. And because the settings are the same, you can use this topic for information about each setting.

For more information, see Creating a certificate installation in Aperture in the *Certificate Management Guide*.

**DID YOU KNOW?** Like most Trust Protection Foundation applications, AWS makes use of the Hostname/Address setting from its parent device object, even though it's a cloud service and not a typical device. This is because AWS is an Internet-based service with a public interface that is the same for all customers. But if you're provisioning to a secondary account, the parent device's Hostname/Address setting in Trust Protection Foundation is actually used to specify the AWS account ID (rather than the expected hostname or address).

This is a confusing but temporary method for managing this specific use case.

For more information about AWS configuration, see "Amazon Web Services (AWS)–Overview" on page 334.

### To create and configure an AWS application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **Amazon Certificate Manager**.
3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.

4. (Optional) In the **Device Certificate** box, click  to select and associate a certificate with the new application.

   > **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

   For detailed information on associating a certificate with an application, see Associating a certificate with an application object.

5. Under **General**, do the following:

   a. In the **Application Name** field, type a name for the new application.

   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.

   c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

      Default system notifications are sent to the contact identities. The default contact is the master administrator.

      > **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
      >
      > Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

   d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

      The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

   You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

   For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Refer to the following table to complete the remaining application object settings:

| Field | Policy | Description |
|---|---|---|
| **Application Information** | | |
| AWS Credential | | Select an Amazon Credential.<br><br>In previous versions of Trust Protection Foundation, you would have selected the access key ID and secret access key credentials. Beginning with version 18.3, you simply create an AWS credential–either a locally hosted or ADFS credential–and then select it here. This allows the support of both locally hosted credentials, and credentials stored externally (ADFS).<br><br>For related information, see Creating and editing an Amazon Credential and Authenticating to multiple AWS accounts using a single Amazon Credential in the *Trust Protection Foundation Administration Guide*. |
| Connection Method | | (Read Only) The **Connection Method** is the protocol that Trust Protection Foundation uses to connect to the server and manage the certificates installed on that server. In an application object's settings, this field is typically read-only. |
| **Installation Settings** | | |
| Certificate Issued By AWS Certificate Manager |  | Choose from the following options:<br><br>◦ **Yes**: Select this option to link the certificate that is enrolled by the Amazon CA to an AWS application.<br><br>This means you'll need to be using the CyberArk Amazon CA template assigned to the certificate.<br><br>**TIP** When you use this setting, the Provision To, Region Code, and IAM Path for Certificate Upload settings are disabled because they're either controlled by the Amazon CA template or are not relevant for other reasons. |

| Field | Policy | Description |
|---|---|---|
| | | ◦ **No**: Select this option and select either ACM or IAM from the Provision To list if the certificates will be issued by another CA. |
| Provision To |  | Either ACM (default) or IAM, depending on where you want to import the certificate. |
| Region |  | If you're provisioning to ACM, choose a region from the list. This should be the region where your AWS application resides.<br><br>When certificates are provisioned to the AWS IAM certificate store, they exist in the primary AWS region, which is us-east-1. If one of those certificates is to be bound to an ELB in a different region, there may be some delay while Amazon replicates the certificate to make it available in other regions. Trust Protection Foundation will attempt to bind the certificate to the ELB for up to 15 minutes to allow time for the AWS replication.<br><br>For more information about AWS regions, visit the following URL:<br><br>http://docs.aws.amazon.com/general/latest/gr/rande.html |
| IAM Path for Certificate Upload (Optional) |  | The IAM certificate store path where the certificate is to be provisioned.<br><br>This value is optional except when provisioning to CloudFront when the path must begin with "/cloudfront/". When left blank, the certificate is provisioned to the root of the IAM certificate store. |
| Replace Existing |  | Select **Yes** if you want to overwrite certificates that were previously provisioned to the ACM store by Trust Protection Foundation. |

| Field | Policy | Description |
|---|---|---|
| | | Enabling Replace Existing lets you automate the certificate renewal lifecycle for ACM-integrated applications like Elastic Beanstalk, API Gateway, and CloudFormation that aren't directly supported by the AWS application driver.<br><br>The Amazon AWS driver supports binding of provisioned certificates to Elastic Load Balancer, Application Load Balancer, and CloudFront. So when you enable the Replace Existing option, Trust Protection Foundation replaces the certificate provisioned to ACM by Trust Protection Foundation with a new certificate.<br><br>The Amazon Resource Name (ARN) used by the previous certificate is used by the new certificate. This means that the previous certificate is not available in ACM for rollback.<br><br>This option is available only when you're provisioning to ACM and when the certificate is not issued by AWS Certificate Manager.<br><br>**IMPORTANT** If you're provisioning a new certificate to an AWS CloudFront distribution to replace an existing certificate in ACM, note that it could take more than an hour before the new certificate becomes active. For perspective, if a certificate isn't replacing an existing one, it usually takes 15 to 20 minutes to become active. |
| Binding Target |  | Indicates to what, if anything, the certificate should be bound.<br><br>◦ No Binding<br>◦ Elastic Load Balancer<br>◦ CloudFront<br>◦ Application Load Balancer |
| **Load Balancer Settings** | | |

| Field | Policy | Description |
|---|---|---|
| Load Balancer Name | | The name of the application or elastic load balancer. |
| Listener Port |  | The TCP port of the load balancer listener to which the certificate will be bound. The default is port 443. |
| Create Listener |  | Select **Yes** if you want to create a listener on the load balancer using the load balancer name and port, and then bind the certificate to it.<br>If you select No, the listener must exist before provisioning. |
| Default Target Group | | The name of the target group associated with the application load balancer listener.<br>This setting doesn't apply to ELB. |
| SNI Certificate | | For Application Load Balancer, enabling the SNI Certificate setting ensures that new certificates are checked for and then added to the SNI list without replacing the default certificate. If setting is not enabled, the application does not check for SNI certificates. |
| Default SNI Certificate | | For Application Load Balancer, enabling the Default SNI Certificate setting ensures that the certificate is added to the SNI list and also replaces the existing default certificate. |
| **CloudFront Settings** | | |

| Field | Policy | Description |
|---|---|---|
| Distribution ID | | The identifier of the CloudFront distribution to which the certificate will be bound. |
| **Certificate Name Settings** | | |
| Certificate Name | | The automatically generated name of the certificate when it was provisioned to the AWS IAM certificate store.<br><br>To ensure uniqueness, the name is derived from parts (or all) of the certificate's common name, expiration date, and serial number.<br><br>This name is retained for two subsequent provisioning cycles of the certificate so that the certificate prior to the one that was just replaced can be automatically removed from the IAM certificate store. |

7. When you're finished, click **Save**.

## What's next?

After you've created an application object, here are other things you can do to manage the new application:

- On the application's **Settings** sub-tab:
  - Click  to push a certificate to its associated application.

    For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.
  - Click  **Reset** to stop processing the application and reset the status and stage.
  - Click  to reattempt installation of the certificate to its associated application, .
  - Click  **Validate Now** to validate the applications associated certificate.

Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

For more information, see "About certificate and application validation" on page 270.

- On the application object's **Validation** tab, you can configure validation settings for the application object.
- On an object's **General** tab:
  - Click the **Log** sub-tab to view any events that are triggered by the template object.
  - Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

## About connecting to AWS service endpoints

For Trust Protection Foundation to communicate with AWS REST endpoints, you first need to understand what information to pass to your firewall team.

The AWS FQDNs with which Trust Protection Foundation needs to communicate vary widely. Endpoint URLs are created based on a specific schema. Amazon documents them by associated services.

The schema Amazon uses for the FQDN is:

```
https://service-code. region-code. amazonaws.com
```

For more information, visit https://docs.aws.amazon.com/general/latest/gr/rande.html.

Depending upon the region and service you're using, as many as 75 FQDNs could be used. Trust Protection Foundation's AWS driver can communicate with the following services:

- ACM
- ALB/ELB
- CloudFront
- IAM

So, if you're using the CyberArk driver to provision to ACM, then you would review the list of 30 or more possible FQDN endpoints here:

https://docs.aws.amazon.com/general/latest/gr/acm.html

For ALB/ELB, there are about 23 FQDNs:

https://docs.aws.amazon.com/general/latest/gr/elb.html

And CloudFront shows just one FQDN:

https://docs.aws.amazon.com/general/latest/gr/cf_region.html

See http://cloudfront.amazonaws.com/.

IAM shows about 25 FQDNs:

https://docs.aws.amazon.com/general/latest/gr/iam-service.html

## Apache configuration

The CyberArk Trust Protection Foundation™ Apache driver supports management of PEM files for Apache web servers running on Linux or Windows. You can also provision the end-entity, chain, and private key to a single file (rather than two or three separate files), and provision certificates that use ECC keys generated centrally or remotely (using the SafeNet Luna SA HSM).

> **DID YOU KNOW?** Using *elliptic curve cryptography* (ECC) as your method of encrypting keys can help you prevent system outages. This is because the ECC algorithm is proving to be more secure than RSA.
>
> Both the CAPI and Apache drivers support provisioning certificates that use elliptic curve cryptography (ECC) keys using central or remote generation. Following provisioning, certificates and keys are extracted and, if using remote generation, download certificates and keys in all supported certificate store formats.
>
> For more information about ECC, see About RSA and elliptic curve cryptography (ECC) key algorithms in the *Certificate Authority and Hosting Platform Integration Guide*.

This section provides the information you need to manage PEM files for Apache web servers.

This chapter contains the following topics:

## Apache driver certificate lifecycle

The following table outlines the stages of the certificate lifecycle for certificates installed on Apache application servers. It also shows you the management levels associated with each lifecycle stage.

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| Stages 0-700 are performed by the Apache application driver only if remote key generation is enabled. If the private key and CSR are locally generated on the Trust Protection Foundation server, stages 0-700 are performed by the X509Certificate Application driver.<br><br>**NOTE** The private key and CSR are remotely generated on the certificate's consumer application (s) if the **Generate Key/CSR on Application** option is enabled in the Certificate object. The **Generate Key/CSR on Application** option is not supported in Apache applications on Windows server environments. | | | | |
| 0 | StartProcessing | No processing. | | |
| 100 | CheckStore | No processing. | | |
| 200 | CreateConfigureStore | No processing. | | |
| 300 | CreateKey | Trust Protection Foundation creates the private key.<br><br>**DID YOU KNOW?** Stage 300 is used for key generation only when the API of a target device separates keypair and CSR generation. When they're combined, both key and CSR generation are always done at stage 400. | x | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 400 | CreateCSR | Trust Protection Foundation creates the Certificate Signing Request (CSR). | x | x |
| 401 | CollectAppGroupData | Trust Protection Foundation collects Application group data. This stage is only applicable when an Entrust nShield Connect HSM HSM is protecting the private key and the certificate is being provisioned to more than one Apache server. | | x |
| 500 | PostCSR | Trust Protection Foundation submits the CSR to the Certificate Authority (CA).<br>**TIP** If you post a manual CSR, this is the first stage of the certificate lifecycle. | x | x |
| 600 | ApproveRequest | Trust Protection Foundation approves the certificate renewal at the CA. | x | x |
| 700 | RetrieveCertificate | Trust Protection Foundation retrieves the certificate from the CA. | x | x |
| 800 | InstallCertificateChain | Trust Protection Foundation installs the root certificate chain file if it is available. | | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| | | If there is an existing chain file, Trust Protection Foundation installs the new certificate chain file only if the **Overwrite Existing Chain** option is selected in the Apache application object.<br><br>**TIP** Root certificates can be discovered and brought under management, or they can be manually imported. For more information, see Managing Root Certificates. | | |
| 801 | InstallPrivateKey | Trust Protection Foundation installs the private key if the private key was not created on the host. | | x |
| 802 | InstallCertificate | Trust Protection Foundation installs the certificate on the Apache application server. | | x |
| 900 | CheckConfiguration | Trust Protection Foundation verifies the configuration after the certificate is installed. | | |
| 1000 | ConfigureApplication | Trust Protection Foundation configures the application to use the installed certificate, if needed. | | |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 1100 | RestartApplication | Trust Protection Foundation restarts the application after the certificate is installed and configured, if needed. | | |
| 1200 | EndProcessing | Trust Protection Foundation completes the certificate processing and, if configured, runs a Validation check on the certificate and application. | | x |
| 1400 | Revocation | Trust Protection Foundation submits a revocation request to the CA.<br><br>**NOTE** Certificate revocation is a certificate operation; it does not involve the application driver. | x | x |

## Permission requirements

The user account you choose to use in Trust Protection Foundation to authenticate with the server so that Trust Protection Foundation can provision certificates or conduct an onboard validation on an Apache application server, must have the following permissions:

- *Read* and *Write* permissions to the directories where Trust Protection Foundation installs the certificate, private key, and root certificate chain files.

  These directories are defined in the Apache application object. For more information, see "Creating an Apache application object" on page 357.

- *Read* and *Write* permissions to the Temp directory defined in the device object.

For more information on the Device object configuration, see "Managing device objects" on page 238.

## Apache prerequisite configuration

To enable Trust Protection Foundation to provision certificates on Apache application servers, you must complete the following procedure.

> **NOTE** If you want to configure remote key generation using an HSM, you must activate Advanced Key Protect, an optional add-on feature to CyberArk Trust Protection Foundation. For more information, see Advanced Key Protect in the *Trust Protection Foundation Administration Guide*.

### To enable Trust Protection Foundation to provision certificates on Apache application servers

1. Configure SSL on the server.

   To use SSL with your Apache web server, you must have ModSSL and OpenSSL installed on your Apache server.

   > **IMPORTANT** Before Trust Protection Foundation can manage certificates on an Apache application server installed on Linux or Windows, you must enable SSL on the server. This procedure can vary, depending on whether or not you want Trust Protection Foundation to provision your certificates. Refer to your Apache, Linux, or Windows documentation for more information.

2. (Conditional) If the Apache application server installed on Linux or Windows has secure virtual hosts, provide an IP address for each.

   Trust Protection Foundation supports both IPv4 or IPv6 connections.

   Because SSL does not support name-based virtual hosts, SSL cannot be configured on name-based virtual hosts unless these virtual hosts use different SSL ports.

   For more information, see your Apache, Linux, or Windows documentation.

3. Grant the following permissions to the user account that Trust Protection Foundation uses to authenticate to the Apache application server on Linux or Windows:

   - Read and Write access to the certificate, private key, and root certificate chain directories. These directories are defined in the Apache Application object. For more information, see "Creating an Apache application object" on page 357.
   - Read and Write access to the temp Directory defined in the Device object. For more information, see "Managing device objects" on page 238.

4. Open the SSH port.

   Trust Protection Foundation uses the Secure Shell (SSH) protocol to manage certificates on an Apache application server installed on Linux or Windows; therefore, Trust Protection Foundation must have access to the web server's SSH port. The default SSH port is port 22.

5. In Policy Tree, create a Device object for the Linux or Windows server where the Apache application is installed.

   For more information, see "Managing device objects" on page 238.

6. In Policy Tree, create and configure an Apache application object for the Apache application server installed on Linux or Windows.

   For more information on creating Application objects, see "Managing application objects" on page 259. For details on the object's settings, see "Creating an Apache application object" on page 357.

7. In Policy Tree, associate the Apache application object with the certificates installed on the Linux or Windows server.

   For more information, see Associating Certificates and Applications.

## About Apache file backups

Whenever Trust Protection Foundation performs a certificate or private key operation, it creates a backup copy of the crypto file in the same directory where the original keystore resides.

If a backup operation fails, Trust Protection Foundation logs an event and halts processing. For example, if a backup operation fails for the JKS driver, the *JKS - Backup KeyStore Failed* event (ID 400E0020) is logged.

> **CAUTION** If a CyberArk driver cannot back up a crypto file, it will not attempt any further certificate and private key operations.

## Creating an Apache Installation

To manage certificate provisioning, you can create an Apache installation, which is similar to the Policy Tree Application object. The Apache installation manages PEM files installed on Apache HTTP servers.

This object provides the information Trust Protection Foundation needs to monitor, enroll, or provision PEM files on its associated Apache or Windows servers, and can even provision the end-entity, chain and private key to a single file (rather than two or three separate files).

When a private key for an Apache certificate is on an Entrust nShield HSM, you can create multiple installations in CyberArk Trust Protection Foundation - Self-Hosted. In the Policy Tree, an Application group appears in the Policy tree for the certificate. Trust Protection Foundation automatically manages the application group.

Before beginning this procedure, make sure that you review the topic, " Apache prerequisite configuration" on page 354.

> **BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

### To create an Apache installation

1. Log in to Certificate Manager - Self-Hosted.
2. Follow the general instructions for Creating a certificate installation in the *Certificate Management Guide*. In the **Add a New Installation wizard**, specify these parameters:
    - **Track, validate, and automate installation of this certificate.** In addition to tracking and validation options above, this option adds the complete end-to-end lifecycle automation of certificates including the ability to have certificates automatically renewed and installed on the host device.
        - **Installation Type**: **Apache HTTP Server**
        - **Yes Configure installation**
3. In the **Installation Settings Wizard**, follow the general instructions in Enabling remote key generation for Apache certificates. For **Private Key Location**, specify **Entrust nShield HSM**.
4. Click **Save and Install**.
5. (Optional) If you want the Apache application servers to share the same certificate and the HSM to manage the private key, repeat these instructions and specify different device parameters. The group generates a new key pair on one server and distributes the key stub and application key token files to the rest of the servers in the farm.

Certificate Manager - Self-Hosted automatically creates an Application group. Each Apache server shares the same certificate.

> **CAUTION** Under the certificate, you can view an Application group. Do not delete the group within the policy tree. Otherwise, results may be unexpected.

## Creating an Apache application object

To enable Trust Protection Foundation to manage PEM files installed on Apache and Windows servers, you must configure the Apache application object. This object provides the information Trust Protection Foundation needs to monitor, enroll, or provision PEM files on its associated Apache or Windows servers, and can even provision the end-entity, chain and–as of Trust Protection Foundation version 18.3– private key to a single file (rather than two or three separate files).

Before beginning this procedure, make sure that you review the topic, " Apache prerequisite configuration" on page 354.

> **BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

> **DID YOU KNOW?** When you add an installation to a certificate, you'll have the option of defining (and editing) this object during that process, which means that you don't have to log in to Policy Tree as the following procedure describes. And because the settings are the same, you can use this topic for information about each setting.
>
> For more information, see Creating a certificate installation in Aperture in the *Certificate Management Guide*.

### To create and configure an Apache application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **Apache**.
3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.

4. (Optional) In the **Device Certificate** box, click  to select and associate a certificate with the new application.

   > **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

   For detailed information on associating a certificate with an application, see Associating a certificate with an application object.

5. Under **General**, do the following:

   a. In the **Application Name** field, type a name for the new application.

   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.

   c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

      Default system notifications are sent to the contact identities. The default contact is the master administrator.

      > **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
      >
      > Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

   d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Under **Application Information**, do the following:

a. Click  next to **Application Credential** to browse for the credential object that you want to use to authenticate with the application.

> **DID YOU KNOW?** Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential might be a user name or private key credential; some drivers–such as F5, which is not SSH-based–can only use the user name credential for authentication.

> **NOTE** The user account you select must have *Read* and *Write* access to the Temporary, Private Key, and Certificate directories.
>
> If you need help with this step, see your system administrator.
>
> For more information, see Working with system credentials in the *Administration Guide*.

> **DID YOU KNOW?** The **Connection Method** is the protocol that Trust Protection Foundation uses to connect to the server and manage the certificates installed on that server. In an application object's settings, this field is typically read-only.

b. (Optional) In the **Port** field, type the port that Trust Protection Foundation should use to communicate with the server where the application is installed.

Trust Protection Foundation uses the SSH protocol to communicate with the application server installed on Linux or Windows. The default SSH port assignment is port 22.

7. Under **Remote Generation Settings**, click the **Private Key Location** list and select where you want remotely generated key pairs to be created and select one of the following options:

**TIP** The Thales SafeNet Luna SA HSM and Entrust nShield HSM settings are not visible if you have not activated Advanced Key Protect. For more information, see Enabling Advanced Key Protect.

- **Device**: Key pairs are generated on the device.

  This is the default setting.

- **Thales SafeNet Luna SA HSM**: Key pairs are generated on a Thales SafeNet Luna SA HSM.

  When you select this option, do the following:

  - In the **Client Tools Path** field, type the directory path where the sautil command from the OpenSSL Toolkit that is located on the device.

    The default path is /usr/safenet/lunaclient/bin.

  - (Required) In the **Partition Password Credential** field, select a password credential that represents the PIN for the HSM partition where the private key is stored.

- **nCipher nShield HSM**: Key pairs are generated on an Entrust nShield HSM.

  When you select this option, do the following:

  - In the **Client Tools Path** field, type the directory path where the generatekey utility and other nShield Core Tools are installed on the device.

    The default path is /opt/nfast/bin.

  - In the **Protection Type** field, select the appropriate level of protection:

    - **Operator Card Set**. Choosing this protection type applies and enables the following:
      - OCS Identifier is the label assigned to the operator card set when it was created
      - OCS Password Credential links to a password credential that has the passphrase assigned to the operator card when the OCS was created.

        If the operator card was created without a passphrase the OCS Password Credential control should be left unassigned.

> **IMPORTANT** Only operator card sets with a quorum of 1 are supported (1-of-N) and the operator card must be inserted into the HSM's card reader slot prior to any interaction by Trust Protection Foundation with the device. *An operator card may be left permanently inserted into the slot; or you can configure a workflow to pause processing until someone has acknowledged reinsertion of the card.*

- **Module**: This is the lowest protection level. It requires that your device has been properly configured to use the HSM for key generation.
- **Softcard**: This is the next highest level of protection. This is a kind of password that is stored on your HSM.

  If you selected Softcard as the Protection Type, then in the **Softcard Identifier** field, enter your softcard's 40-character hash.

  This option requires that the device is properly configured to use the HSM for key generation and that a softcard has been previously generated using the HSM, and that the requester knows the passphrase for that softcard.

  > **NOTE** If you set Reuse Private Key for Service Generated CSRs to Yes on a certificate's policy, the protection type is ignored because it cannot be changed for an existing private key. If you need to change the protection type, you must set **Reuse Private Key for Service Generated CSRs** to **No**.
  >
  > For more information, see "Configuring HSM-based remote key generation" on page 273 and Setting policy on a folder (Aperture).

- **Private Key Alias**: Displays the HSM key alias value for this key using the Private Key Label Apache object attribute (to save the data).

  While this read-only field is visible on every Apache application object but is enabled for the Entrust nShield HSM option only.

  The private key alias is created by combining the time-stamp (YYMMDDhhmmss) and file name of the key (without the file extension). For example, *190316142039_MyPrivateKey*.

8. Refer to the following table to complete the remaining settings:

| Field | Policy | Description |
|---|---|---|
| **Apache Settings** | | The following are server-specific certificate settings. They are referenced only when you associate a certificate with the current Apache application object. |
| Private Key File |  | The path and filename on the Apache application server where Trust Protection Foundation installs the private key.<br>This setting must match the Apache application's private key file configuration. For more information, refer to your Apache documentation. |
| Private Key Credential |  | The credential required to access the private key file for certificate renewal.<br>**To select a private key password credential**<br>a. Click  to open the **Credential Selector** dialog. |

| Field | Policy | Description |
|---|---|---|
| | | b. Select the credential required to access the private key file for certificate renewal, and then click **Select**.<br><br>For more information, see Managing system credentials in the *CyberArk Trust Protection Foundation Administration Guide*.<br><br>Trust Protection Foundation does not include the private key password on the command line when performing key management operations. Instead, it provides the password when prompted. |
| Certificate File |  | The path and filename on the Apache application server where Trust Protection Foundation installs the certificate. |

| Field | Policy | Description |
|---|---|---|
| | | This setting must match the Apache application's certificate file configuration. For more information, refer to your Apache documentation.<br><br>**DID YOU KNOW?** If you want to provision the end-entity, chain and private key to a single file (rather than two or three separate files), then specify the same path and filename for the Certificate File and Certificate Chain File settings, and/or Private Key File settings on the application object. |
| Certificate Chain File |  | The path and filename on the Apache application server where Trust Protection Foundation writes root certificates.<br><br>This setting must match the Apache application's certificate chain file configuration. For more information, refer to your Apache documentation. |

| Field | Policy | Description |
|---|---|---|
| Overwrite Existing Chain |  | Overwrites the existing certificate chain file when Trust Protection Foundation installs a new certificate and private key. |

9. (Optional) Under **File Ownership and Permissions**, select **Yes** on the **Set Owner and Permissions after Provisioning Files** drop-down—if you want to set specific permissions and ownership on files after they have been provisioned by Trust Protection Foundation—and then do the following:

    a. In the **Owner** field, type the user account name of the user who should have access to the provisioned files.

    > **BEST PRACTICE** Who you assign as owners and approvers of your certificates is an important part of your PKI strategy. This is especially true because employees continue to pose the greatest threat to securing trust. Typically, this is because many employees fail to follow security best practices.

    b. From the **Owner Permissions** list, select the level of permissions you want to grant to the owner (*Read*, or *Read and Write*).

    c. In the **Group** field, type the group name to which the owner belongs.

    d. From the **Group Permissions** list, select the level of permissions you want to grant to that group (*None*, *Read*, or *Read and Write*).

10. When you are finished, click **Save**.

For additional HSM configuration information, see "Managing applications using HSM-protected keys and Advanced Key Protect" on page 271 and "Configuring HSM-based remote key generation" on page 273.

## What's next?

After you've created an application object, here are other things you can do to manage the new application:

- On the application's **Settings** sub-tab:
  - Click  to push a certificate to its associated application.

    For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.
  - Click  **Reset** to stop processing the application and reset the status and stage.
  - Click  to reattempt installation of the certificate to its associated application, .
  - Click  **Validate Now** to validate the applications associated certificate.

    Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

    For more information, see "About certificate and application validation" on page 270.
- On the application object's **Validation** tab, you can configure validation settings for the application object.
- On an object's **General** tab:
  - Click the **Log** sub-tab to view any events that are triggered by the template object.
  - Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

## Azure Key Vault configuration

Microsoft Azure is a collection of integrated cloud services that developers and IT professionals use to build, deploy, and manage applications through Microsoft's global network of data centers. When Azure is integrated with Trust Protection Foundation, certificates, keys, and chains can be stored in Azure Key Vault, allowing for easier and faster provisioning of certificates.

In order to create and configure an integration between Microsoft Azure and Trust Protection Foundation, you must first create and configure an Azure portal. Visit https://azure.microsoft.com/en-us/free/ to learn how.

This section provides the information you need to configure Trust Protection Foundation to manage certificates stored in Azure Key Vault.

## Using Microsoft Azure with Trust Protection Foundation

To integrate Trust Protection Foundation with Azure, you need to complete prerequisite steps in both Microsoft Azure and Trust Protection Foundation.

### In Microsoft Azure

Before configuring Trust Protection Foundation for integration with Azure, you must perform a series of critical prerequisite tasks in Azure. See "Preparing Azure for integration with CyberArk" below.

> **DID YOU KNOW?** You can also perform many of the prerequisite tasks required for interoperability between Trust Protection Foundation and Microsoft Azure by using PowerShell cmdlets. For more information, see " Using PowerShell to configure Microsoft Azure for integration with Trust Protection Foundation" on page 369

### In CyberArk Trust Protection Foundation

After completing the Azure configuration prerequisites, perform the following tasks in Trust Protection Foundation:

- Configure Trust Protection Foundation to connect to the Azure Key Vault without having to log in and present a certificate each time.

  See "Setting up authentication to Microsoft Azure from Trust Protection Foundation" on page 371.

- Create and configure the required objects in Trust Protection Foundationso that it can store certificates, private keys, and chains in your Azure Key Vault.

  See "Creating a Microsoft Azure device object" on page 373 and "Creating an Azure Key Vault application object " on page 374.

- Create an Azure Key Vault application object in Trust Protection Foundation.

  See "Creating an Azure Key Vault application object " on page 374.

### Preparing Azure for integration with CyberArk

Before you configure Trust Protection Foundation for integration with Microsoft Azure, perform the following prerequisite tasks.

**IMPORTANT** Because Microsoft continues to update the Azure user interface, please refer to the most current Microsoft Azure documentation for specific details.

1. Log in to your Azure portal.
2. (Conditional) If you've not already done so, create a key vault, unless you plan to do so using another method, such as PowerShell cmdlets, from a command line, or using an API.
3. Go to your Azure Active Directory and create a new Application Registration.
4. From the new Application Registration, import the certificate to that Application.

   The same certificate will have to be imported into Trust Protection Foundation and used for authentication when provisioning to Azure Key Vault.
5. Grant Azure Key Vault API permissions to the Application.
6. For Azure role-based access control (RBAC) Key Vaults:
   a. The Key Vault Certificates Officer and Key Vault Secret Officer roles must be granted to the Application Registration.
   b. The Key Vault Secrets User role must granted to the Microsoft Azure App Service for the Key Vault.
7. For Vault Access Policy Key Vaults:
   a. From the Access Policies blade of the Key Vault, add the newly created Application and Microsoft Azure App Service as principals.
   b. From the Access Control (IAM) blade of the Key Vault, assign Contributor and Key Vault Contributor roles to the newly created application.
8. From the Access Control (IAM) blade of the Key Vault, assign Contributor and Key Vault Contributor roles to the newly created application.

**NOTE** Before moving on to the Trust Protection Foundation configuration steps, wait approximately 10 to 15 minutes for the Azure changes to take effect.

## Azure permission requirements

While there are multiple ways to configure and meet permissions requirements, to successfully provision certificates to Azure Key Vault and Web Application, the following is recommended:

- Certificate key/pair must be present
- Application Registration must be created (which is the "user", also referred to as Application ID)
- Azure Key Vault and Azure Service Management API permissions must be granted to the Application Registration
- The Contributor Role must be granted to the Application Registration in the Service Plan resource
- The Contributor Role must be granted to the Application Registration in the Resource Group to which the Web Application belongs
- For Azure role-based access control (RBAC) Key Vaults:
    - The Key Vault Certificates Officer and Key Vault Secrets Officer roles must be granted to the Application Registration for the Key Vault
    - The Key Vault Secrets User role must be granted to the Microsoft Azure App Service for the Key Vault
- For Vault access policy Key Vaults:
    - The Contributor role and Key Vault Contributor role must be granted to the Application Registration for the Key Vault
    - The Application Registration and Microsoft Azure App Service must be added as principals to the Key, Secret and Certificate Management policy in the Access Policy of the Key Vault

**TIP** When integrating Azure with Trust Protection Foundation, you might encounter the following error message:

Error: Failed to bind a key vault certificate to the web application. No default Subscription has been designated. Check your Azure account Subscriptions and select a default one.

This error typically occurs when you use an Application ID / Principal that does not have the proper permissions set. Verify the permissions settings and try again.

## Using PowerShell to configure Microsoft Azure for integration with Trust Protection Foundation

**IMPORTANT** Microsoft Azure Powershell module is considered deprecated and support removed in February 2024. See https://docs.microsoft.com/en-us/powershell/module/azurerm.keyvault/.

Although optional, many of the prerequisite tasks required for interoperability between Trust Protection Foundation and Microsoft Azure can be performed more effectively using PowerShell cmdlets.

> **TIP** For a complete list of PowerShell Key Vault cmdlets, visit https://docs.microsoft.com/en-us/powershell/module/azurerm.keyvault/.

Here are some of the more common PowerShell cmdlets you can use:

| Action | Commands and examples |
|---|---|
| To create a key vault for storing and protecting certificates and keys in Azure. | **New-AzureRmKeyVault**<br><br>Example:<br><br>`New-AzureRmKeyVault -VaultName "ExampleKeyVault" -ResourceGroupName "ExampleResourceGroup" -Location "West US"` |
| Create an Application Registration for Trust Protection Foundation in your Azure Active Directory | **New-AzureRmADApplication**<br><br>Example:<br><br>`$cer = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2`<br>`$cer.Import("C:\Temp\AzureCredential.crt")`<br><br>`$guid = [guid]::NewGuid().Guid`<br><br>`Import-Module AzureRM.Resources`<br><br>`$cred = New-Object Microsoft.Azure.Commands.Resources.Models.ActiveDirectory.PSADKeyCredential`<br>`$cred.KeyId = $guid`<br>`$cred.CertValue = [System.Convert]::ToBase64String($cer.GetRawCertData())`<br>`$cred.StartDate = $cer.NotBefore`<br>`$cred.EndDate= $cer.NotAfter`<br><br>`New-AzureRmADApplication -DisplayName "VenafiTPP" -HomePage "http://venafi-tpp" -IdentifierUris "http://$guid" -KeyCredentials $cred` |
| Create a Service Principal for the Application Registration | **New-AzureRMADServicePrincipal**<br><br>Example: |

| Action | Commands and examples |
|---|---|
| | New-AzureRMADServicePrincipal -ApplicationId "1e8ed931-3cfc-466d-ac05-e02bd7a0132b" |
| Grant the Service Principal access to the Key Vault | **Set-AzureRmKeyVaultAccessPolicy**<br>Example:<br>Set-AzureRmKeyVaultAccessPolicy -VaultName "ExampleKeyVault" -ServicePrincipalName "2d66f61a-c3d5-450a-a747-af2659165734" -PermissionsToCertificates get,list,create,import |

Some Azure Key Vault administrative tasks, such as deletion, are not yet supported by the Azure web portal. So the following additional PowerShell cmdlets might be helpful:

| Action | Commands and examples |
|---|---|
| Retrieve certificates from the Key Vault with the option to include past versions | **Get-AzureKeyVaultCertificate**<br>Example:<br>Get-AzureKeyVaultCertificate -VaultName "ExampleKeyVault" -Name "webapp-company-com" |
| Delete a certificate from the Key Vault | **Remove-AzureKeyVaultCertificate**<br>Example:<br>Remove-AzureKeyVaultCertificate -VaultName "ExampleKeyVault" -Name "webapp-company-com" |

## Setting up authentication to Microsoft Azure from Trust Protection Foundation

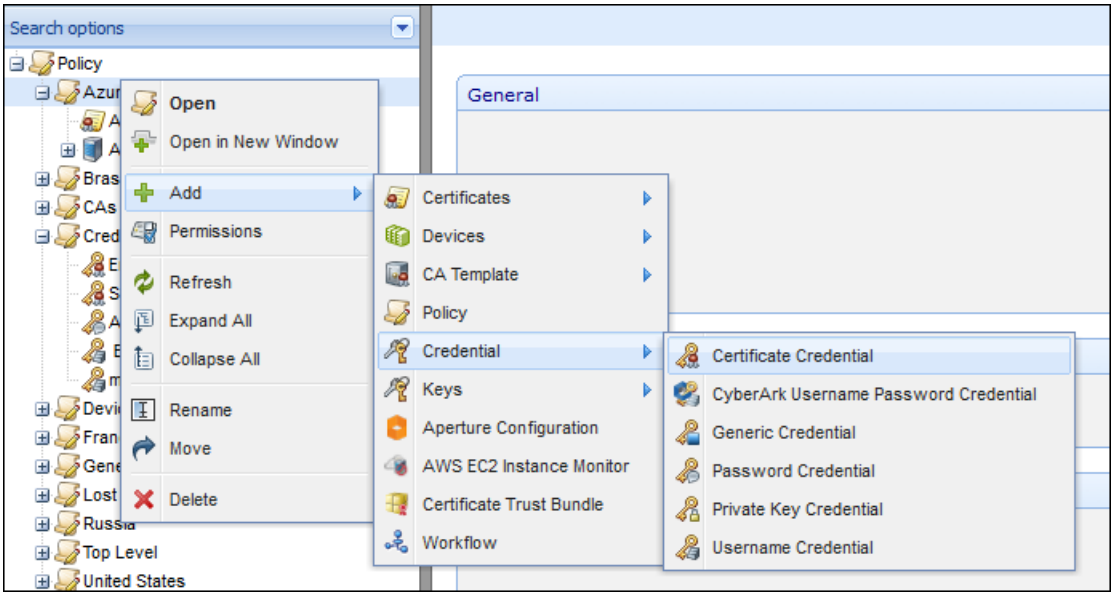
In order to connect to the Azure Key Vault without having to log in and present a certificate each time, you can create and enroll a certificate and create an associated credential for it. The data is then stored in Trust Protection Foundation and presented to the Azure Key Vault whenever you access it.

### To create a credential object for authenticating with Azure Key Vault

1. In the **Policy** tree, create and enroll a certificate that you'll use to log in to the Azure Key Vault.

   If you're not sure how to do that, see Creating a new certificate.

2. Highlight the root Policy, then select **Add** > **Credential** > **Certificate Credential**.




3. Click **Certificate Credential.**

4. Under **Certificate Credential**, type the **Credential Name**, **Description**, and **Contact(s)**.

5. Under **Certificate Data**, select **Link to existing certificate**, and then choose the certificate you created in Step 1.

### On the Azure side, configure the certificate to be used by Trust Protection Foundation.

1. In Azure Navigate to the **Application Registration** and select **Certificate & Secrets**.

2. To upload your certificate, select **Upload certificate**.

Now the certificate can authenticate Trust Protection Foundation to Azure.

## Creating a Microsoft Azure device object

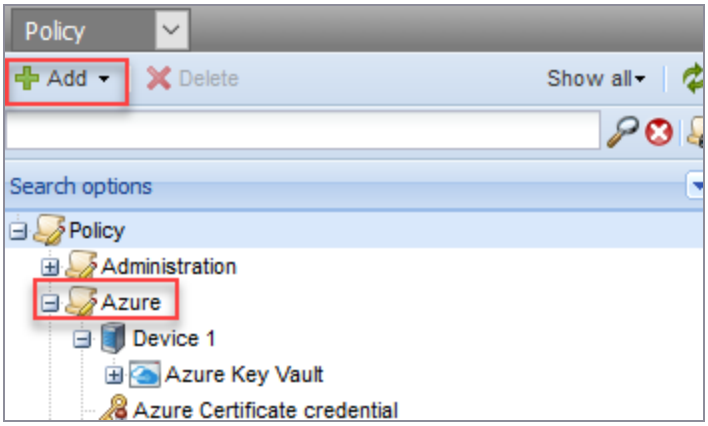
Trust Protection Foundationrequires a policy, device, and Azure Key Vault application object in order to store certificates, private keys, and chains in the Azure Key Vault.

**DID YOU KNOW?** When you add an installation to a certificate, you'll have the option of defining (and editing) this object during that process, which means that you don't have to log in to Policy Tree as the following procedure describes. And because the settings are the same, you can use this topic for information about each setting.

For more information, see Creating a certificate installation in Aperture in the *Certificate Management Guide*.

### To create and configure the Azure device object

1. In the Policy Tree, add a policy named *Azure* to the **Policy** tree.




2. Right-click the Azure policy you just created, point to **Add** > **Devices**, and then click > **Device**.
3. Under **General**, type a name for your device in the **Device Name** field, and if you want, type a brief description of the purpose of the device in the **Description** .

   **NOTE** This device is simply a placeholder that represents the Azure cloud service provider. No other configuration is necessary.

4. Click**Save**.

## Creating an Azure Key Vault application object

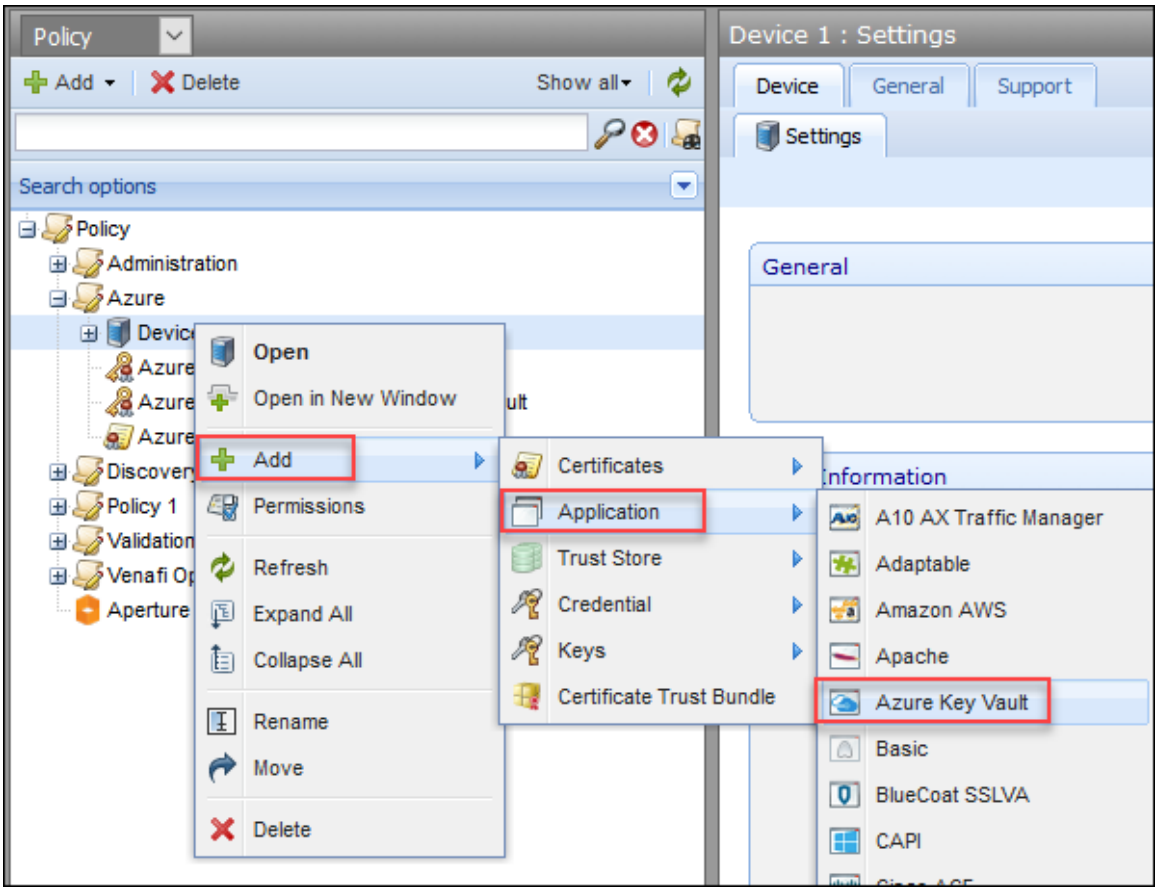
To store certificates, an Azure Key Vault application object must be created in Trust Protection Foundation. Then, in Azure, the Service Principal must be granted the required permissions to manage certificates.

You can also configure Trust Protection Foundation to bind your certificates to web applications automatically during provisioning so that you don't have to create the binding using other methods.

> **TIP** If you plan to bind certificates to a web application, make sure that you've set up your web application already before you complete the following procedure.

### To create an Azure Key Vault application

1. In Policy Tree, click the Azure policy that you created earlier.
2. Find the Azure device that you created under the Azure policy.
3. Highlight the device and then right-click it.
4. Choose **Add** > **Application** > **Azure Key Vault**.




5. Under **Certificate**, for **Associated Certificate**, click ... to find the certificate that you want to associate with the Azure Key Vault application.

> **NOTE** You can enable only one associated application per remote key generation. To see the differences between the two methods of private key generation, see *Remote vs. central key generation*.
>
> To enable remote key generation:
>
> a. Click the certificate name to open the certificate.
>
> b. In the certificate's **Settings** tab, under **CSR Handling**, set **Generate Key/CSR on Application** to Yes.

6. In **Application Information**, do the following:

a. In the **Application ID** field, type the application ID of the Azure Service Principal.

b. Click **Certificate Credential** to select the required credential.

7. In **Installation Settings**, do the following:
   a. In **Azure Key Vault Name**, type the name of your key vault.

   b. In the **Certificate Name** field, type the name of the certificate that is going to be used for tracking the certificate in the Azure Key Vault.

   > **NOTE** The Certificate Name is automatically generated from Certificate Common Name (or DNS SANs if you don't specify a name).
   >
   > If you do specify a name, dots are changed to hyphens.
   >
   > Example: *dsmith.denver.sales* is converted to *dsmith-denver-sales*. Also, special characters are not allowed in the name and are removed.

   > **TIP** When you provision a new certificate to Azure Key Vault using a certificate name that is already in use, Azure automatically updates any existing bindings to that certificate name. However, they won't likely be activated as quickly as the bindings that are configured to be updated using the driver.

   c. (Conditional) If you want Trust Protection Foundation to extract certificates and private keys from Azure Key Vault, set **Private Key Exportable** to **Yes**.

   d. Use **Chain Order** to select the order of the certificate chain, with options such as **Root first** and **End-entity first**.

8. (Conditional) If you want to bind certificates to web applications, then in the **SSL Settings** box, do the following:
   a. Set **Bind Certificate to Web Application** to **Yes**.

   b. In the **Web Application Name** field, type the name of the web application to which you want to bind certificates.

   c. (Optional) A subscription ID is only required if the web application to be bound is in a different subscription than the specified key vault. Enter the subscription ID of the web application.

   d. If a binding doesn't exist already, set **Create New Binding** to **Yes**.

e. Set **Create/Update binding for each DNS SAN** to **No** if you only want to create/update a specific FQDN.

   If you select No here, you'll enter a single hostname in the Specify Binding Hostename(s) field (described below). Select Yes if you want Trust Protection Foundation to automatically create/update one binding for every DNS SAN in the certificate.

f. Click the **SSL Type** field and select whether to use **SNI** or **IP Based**.

g. In **Specify Binding Hostname(s)** , type one or more hostnames.

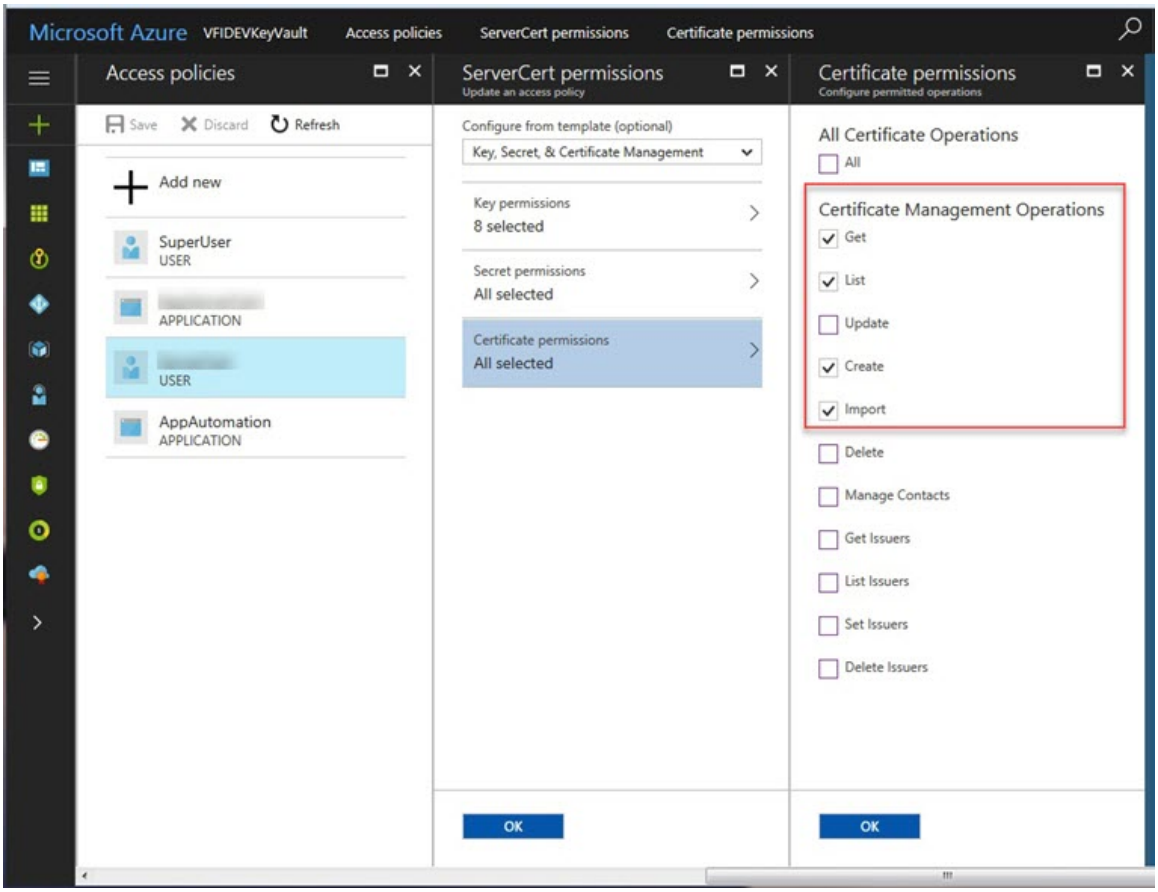
9. When you're done, click **Save**.

### Create an Azure Active Directory

You'll need to create an Azure Active Directory that includes *App Registration* and *Service Principal*. Refer to Azure's product documentation for instructions.

### Permissions required for the Service Principal in the Azure Key Vault

In the Azure Key Vault, the Service Principal must have the *Get*, *List*, *Import*, and *Create* certificate permissions.




## Adding an installation to a certificate using Azure Key Vault

You can add Azure Key Vault to an existing certificate. This allows Azure Key Vault to provide credentials when it's time for the certificate to be provisioned.

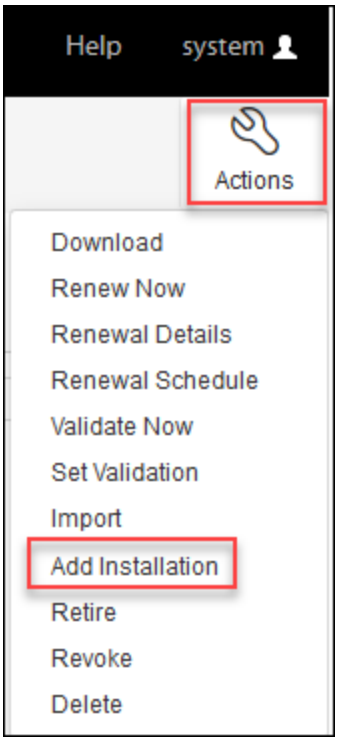
### To add Azure Key Vault to an existing certificate

1. Find the certificate to which you want to add an installation (application).

   To learn how to use filters to find a certificate, see *Finding assets in Aperture using filters*.

2. In the side bar, click **Installations**.

3. Click **Actions** > **Add Installation**.




4. In **Add New Installation**, select **Track, validate, and automate installation of this certificate**.
5. Click **Next**.
6. Under **Find Existing Device**, select the appropriate device from the list.
7. Under **Installation Type**, select **Azure Key Vault**.
8. Click **Add Installation**.

   In **Installation Added**, you can configure settings that allow the certificate to be automatically installed.

   For more information about adding an installation, see *Creating an Certificate Installation in Aperture*.

## Basic application configuration

The Basic application is a limited-functionality application object. It is created under a Device object to represent an unmanaged application. In function, the Basic Application object allows administrators to track unmanaged applications where certificates are installed and route notifications to the application contacts.

When Trust Protection Foundation runs a network validation on a certificate associated with a Basic Application object, it validates that the certificate is available at the address/port configured in the Basic Application object.

> **DID YOU KNOW?** Basic applications are not applicable to provisioning. Therefore, Trust Protection Foundation skips them when the Management Type of the certificate is set to Provisioning and processing reaches the provisioning stage.

This section provides the information you need to configure a Basic Application object to associate a monitored certificate with a Device object.

This chapter contains the following topics:

## Permission requirements

There are no permission requirements for the Basic application object.

## Basic application configuration checklist

To enable CyberArk Trust Protection Foundation™ to associate monitored certificates with device objects, you must complete the following high-level tasks:

- In the Policy Tree, create a device object for the server you want to associate with the certificate.

  For more information, see "Managing device objects" on page 238.

- In the Policy Tree, create and configure a Basic Application object under the Device object.

  For more information on creating Application objects, see "Managing application objects" on page 259. For details on the object's settings, see "Creating a Basic application object" below.

- In the Policy Tree, associate the Basic Application object with the certificate.

  For more information, see Associating Certificates and Applications.

## Creating a Basic application object

The Basic application object allows administrators to track unmanaged applications where certificates are installed and route notifications to the application contacts. When Trust Protection Foundation runs a network validation on a certificate associated with a Basic application object, it validates that the certificate is available at the address/port configured in the Basic application object.

**BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

### To create and configure a Basic application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **Basic**.

3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.

4. (Optional) In the **Device Certificate** box, click ... to select and associate a certificate with the new application.

   > **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

   For detailed information on associating a certificate with an application, see Associating a certificate with an application object.

5. Under **General**, do the following:

   a. In the **Application Name** field, type a name for the new application.

   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.

   c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

      Default system notifications are sent to the contact identities. The default contact is the master administrator.

      > **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
      >
      > Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

   The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

   You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

   For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. When you are finished making changes, click **Save**.

## What's next?

After you've created an application object, here are other things you can do to manage the new application:

- On the application's **Settings** sub-tab:
  - Click  to push a certificate to its associated application.

    For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.

  - Click  **Reset** to stop processing the application and reset the status and stage.
  - Click  to reattempt installation of the certificate to its associated application, .
  - Click  **Validate Now** to validate the applications associated certificate.

    Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

    For more information, see "About certificate and application validation" on page 270.

- On the application object's **Validation** tab, you can configure validation settings for the application object.

- On an object's **General** tab:
  - Click the **Log** sub-tab to view any events that are triggered by the template object.
  - Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

## CAPI driver (and IIS 7.x) configuration

The *cryptographic application programming interface* (CAPI) application driver installs certificates and private keys in the CAPI store of a Windows host running Windows Remote Management (WinRM).

The CAPI driver relies on native Windows functionality, fully supported by Microsoft, and has minimal port requirements, which greatly simplifies its integration in even the largest organizations.

To help you prevent system outages, you could also choose to use *elliptic curve cryptography* (ECC) as your method of encrypting keys. Both the CAPI and Apache drivers support provisioning certificates that use elliptic curve cryptography (ECC) keys using central or remote generation. Following provisioning, certificates and keys are extracted and, if using remote generation, download certificates and keys in all supported certificate store formats.

For more information about ECC, see About RSA and elliptic curve cryptography (ECC) key algorithms in the *Certificate Authority and Hosting Platform Integration Guide*.

Although optional, the CAPI driver can bind the installed certificates and private keys to an IIS web site. Therefore, it is the best choice for provisioning certificates to IIS 7.0 and later.

**DID YOU KNOW?** As Microsoft Windows® evolved from a consumer to an enterprise operating system, Microsoft introduced a cryptographic application programming interface (CryptoAPI) built around an operating system-centric certificate and key store commonly referred to as the CAPI store.

Today, nearly all Microsoft (and some third-party) applications that use certificates access them from the CAPI store. Server-based applications such as IIS®, Exchange®, SQL Server®, and Lync® are all designed to work with certificates in the Local Computer\Personal CAPI store.

In 2006, Microsoft released a new scripting language called PowerShell. But it wasn't until the release of Windows Management Framework 2.0 that Windows had a native feature analogous to SSH. WinRM introduced the ability to securely execute commands on remote Windows systems. PowerShell provides robust support for manipulating certificates in the CAPI store and for configuring IIS. Together, WinRM and PowerShell form the foundation of the CAPI driver included with Trust Protection Foundation.

**NOTE** For information about the trust store, see About creating and configuring trust stores in the *CyberArk Trust Protection Foundation Certificate Management Guide*.

This section provides the information you need to correctly configure a CAPI driver object and contains the following sections:

This chapter contains the following topics:

## CAPI driver certificate lifecycle

The following table outlines the stages of the certificate lifecycle for certificates installed to the CAPI store and identifies the management level associated with each stage.

> Lifecycle stages 800 through 1200 are handled by application drivers. However, not all stages are defined for every driver. For more information about lifecycle stages, see "About certificate lifecycle management" on page 42 in the *CyberArk Trust Protection Foundation Certificate Management Guide*.

For the CAPI driver, stages 800 through 804, are defined in the following table.

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 800 | InstallCertificateChain | Ensures that each of the applicable issuer certificates has been installed in the CAPI store. | | x |
| 801 | InstallCertificate | Installs the certificate and private key into the Local Computer\Personal CAPI store. | | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 802 | GrantPrivateKeyAccess | Grants the Private Key Trustee read access to the key if configured to do so. | | x |
| 803 | BindCertificateToWebSite | Binds the certificate to an IIS Web Site if configured to do so | | x |
| 804 | RemoveObsoleteCertificates | Removes the previous version of a certificate that has been provisioned by Trust Protection Foundation two or more times.<br><br>This stage helps to limit the number of versions of a certificate that are kept in the CAPI store to the most recent version, plus one version back.<br><br>**EXAMPLE** Suppose you have provisioned a certificate three times. Each time, Trust Protection Foundation created a new version of the certificate in case of the unlikely event that you needed to manually restore an earlier version. When Trust Protection Foundation created the third version, it automatically deleted the first version, leaving only the most recent and previous versions in the CAPI store.<br><br>For more information about how this works, see About Generational Management in the Trust Protection Foundation Administration Guide. | | |

## CAPI driver permissions requirements

Trust Protection Foundation requires a Windows administration account whenever it provisions certificates or conducts an onboard validation. Therefore, verify that the user account Trust Protection Foundation uses to authenticate to the Windows server is a Windows administrator.

# CAPI driver prerequisite configuration

The CAPI application driver installs certificates and private keys in the CAPI store of a Windows host running WinRM. You can also install CAPI certificates using the Server Agent. For more information about Server Agent, see Server Agent–Introduction in the *Server Agent Installation and Configuration Guide*.

**IMPORTANT** If you want to configure remote key generation using an HSM, you must activate Advanced Key Protect, an optional add-on feature to CyberArk Trust Protection Foundation. For more information, see Advanced Key Protect in the *Trust Protection Foundation Administration Guide*.

This topic outlines device and Trust Protection Foundation system requirements.

## Target device requirements

- For Windows Server 2016, Windows Management Framework (WMF) 5.1 or higher is required.
- Windows Remoting (WinRM) must be enabled.

  Disabled by default, it can be enabled by executing one of the following on the target system (a computer certificate is required to enable WinRM over HTTPS):

  ```
  winrm quickconfig
  winrm quickconfig -transport:https
  ```

  Trust Protection Foundation tries to connect using the port number specified by the CAPI application object. Typically, HTTPS is attempted before HTTP; the order is only reversed when the port specified is 5985 or 80, which are the standard HTTP ports for WinRM.

  WinRM over HTTP *only* allows Kerberos authentication without additional configuration. In order to allow WinRM over HTTP to accept NTLM for authentication, the WinRM client on the Trust Protection Foundation server must have the target system in its Trusted Host list, which means that system will be trusted even though its identity cannot be authenticated.

  **CAUTION** NTLM and NTLMv2 authentication is vulnerable to a variety of malicious attacks, including SMB replay, man-in-the-middle, and brute force attacks.

  Assuming that this risk is understood and acceptable, accounting for each target individually might not be feasible; so the following command can be executed on the Trust Protection Foundation server to allow it to use NTLM when authenticating with WinRM over HTTP:

```
winrm set winrm/config/client '@{TrustedHosts="*"}'
```

One or more inbound rules may need to be added if the Windows Firewall is active on the target system. These rules would only need to allow access from each of the Trust Protection Foundation servers that are capable of provisioning to the target device on the WinRM listening port.

An inbound rule might need to be added if the Windows Firewall is active on the target system. This can be accomplished using the MMC or by executing the following command:

```
netsh advfirewall firewall add rule name="Windows Remote Management (HTTPS-In)" dir=in
action=allow enable=yes profile=any remoteip=<Trust Protection Foundation IP>=5986
protocol=tcp program=System
```

- (Required) Windows PowerShell Snap-In for IIS 7.0
- User account with permissions to access the server via WinRM, execute PowerShell commands that import and export certificates and private keys from the Windows CAPI store, and that can modify IIS configurations.

## Trust Protection Foundation system requirements

- Windows Management Framework (WMF) 3.0 or higher

## CAPI Connectivity

The following table summarizes CAPI connectivity options and settings.

| | Trusted Domain/Forest (Supports Kerberos Authentication) | Non-Trusted Domain/Forest (Supports NTLM Authentication Only) |
|---|---|---|
| **HTTP (5985/tcp)** | No additional configuration necessary. | WinRM on the Trust Protection Foundation systems is configured to explicitly consider which target systems are trusted<br><br>For example, winrm set winrm/config/client @ {TrustedHosts="*"} would allow the trusting of every target for NTLM. |
| **HTTPS (5986/tcp)** | WinRM on the target system must have (and be using) an SSL certificate that is trusted by the Trust Protection Foundation server.<br><br>This trust requirement setting can be globally disabled by creating a registry DWORD value in the HKLM\Software\Venafi\Platform key directory called WinRMSkipCACheck=1. | WinRM on the target system must have (and be using) an SSL certificate that is trusted by the Trust Protection Foundation server.<br><br>This trust requirement setting can be globally disabled by creating a registry DWORD value in the HKLM\Software\Venafi\Platform key directory called WinRMSkipCACheck=1. |

## Creating a CAPI application object

This section describes the configuration settings needed to enable Trust Protection Foundation to install an SSL certificate key pair into the CAPI store of a Microsoft Windows host.

**BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

**DID YOU KNOW?** When you add an installation to a certificate, you'll have the option of defining (and editing) this object during that process, which means that you don't have to log in to Policy Tree as the following procedure describes. And because the settings are the same, you can use this topic for information about each setting.

For more information, see Creating a certificate installation in Aperture in the *Certificate Management Guide*.

### To create and configure a CAPI application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **CAPI**.
3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.
4. (Optional) In the **Device Certificate** box, click ... to select and associate a certificate with the new application.

   **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

   For detailed information on associating a certificate with an application, see Associating a certificate with an application object.
5. Under **General**, do the following:
   a. In the **Application Name** field, type a name for the new application.
   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.

c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

Default system notifications are sent to the contact identities. The default contact is the master administrator.

> **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
>
> Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Under **Application Information**, do the following:

a. Click  next to **Application Credential** to browse for the credential object that you want to use to authenticate with the application.

> **DID YOU KNOW?** Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential might be a user name or private key credential; some drivers–such as F5, which is not SSH-based–can only use the user name credential for authentication.

> **NOTE** The user account you select must have *Read* and *Write* access to the Temporary, Private Key, and Certificate directories.
>
> If you need help with this step, see your system administrator.
>
> For more information, see Working with system credentials in the *Administration Guide*.

> **DID YOU KNOW?** The **Connection Method** is the protocol that Trust Protection Foundation uses to connect to the server and manage the certificates installed on that server. In an application object's settings, this field is typically read-only.

b. In the WinRM Port, specify the port number that Trust Protection Foundation should use to communicate with CAPI devices.

   The default WinRM ports assignments are port 5985 and 5986 for HTTP and HTTPS, respectively.

7. Under **Remote Generation Settings**, click the **Private Key Location** list and select where you want remotely generated key pairs to be created:

> **TIP** The Thales SafeNet Luna SA HSM and Entrust nShield HSM settings are not visible if you have not activated Advanced Key Protect. For more information, see Enabling Advanced Key Protect.

   - **Device**: Select this option to generate key pairs on the device.

     This is the default setting.

   - **Thales SafeNet Luna SA HSM**: Select this option to generate key pairs on a Thales SafeNet Luna SA HSM.

     - In the **Key Label** field (that appears only after you select Thales SafeNet Luna SA HSM), type the label that you want used for keys created by the CAPI driver in your HSM.

       This option makes it easier for you to identify which certificates correspond to the newly generated keys. If you leave this field empty, Windows automatically generates a unique name whenever a new key pair is created. Alternatively, you could choose to change the key label manually before renewing the certificate.

> **DID YOU KNOW?** Trust Protection Foundation uses the label you specify each time that it provisions the certificate. Because Microsoft Windows allows only one instance of a particular key label, you must set **Reuse Private Key for Service Generated CSRs** to **Yes** on the certificate's policy, which prevents a possible error the first time the certificate is renewed. For more information, see Policy object certificate settings and Setting policy on a folder (Aperture).

- **Entrust nShield HSM**: Select this option to generate key pairs on an Entrust nShield HSM using Module protection mode.

8. Refer to the following table to complete the remaining settings:

## CAPI Settings

| Field | Policy | Description |
|---|---|---|
| Store Name | Yes | You can specify which type of CAPI Store to use when provisioning certificates, namely Personal, Web Hosting, and Remote Desktop. |
| Friendly Name | No | The friendly name that identifies the certificate in the Windows CAPI store (required).<br><br>Windows does not require the Friendly Name to be unique. If a name collision is encountered during validation or extraction, Trust Protection Foundation uses the most recently issued certificate.<br><br>**NOTE** When certificates are discovered by the Server Agent, the agent captures and sends the certificate label to Trust Protection Foundation using the existing friendly name JSON object property. Trust Protection Foundation then sets the certificate label value when it creates the new application object. |

| Field | Policy | Description |
|---|---|---|
| | | **DID YOU KNOW?** Depending on the trust store, this setting goes by different names. For example, GSK refers to this setting as the *Certificate Label*, JKS refers to it as the *Certificate Alias*, and CAPI and PKCS#12 refer to it as the *Friendly Name*. |
| Exportable | Yes | Controls whether the private key installed by Trust Protection Foundation is exportable from the CAPI store.<br><br>**NOTE** When configuring CAPI for use with remote generation using an HSM and you have **Generate Key/CSR on Application** set to **Yes**, **Private Key Location** set to **Thales SafeNet Luna SA HSM**, and **Reuse Private Key for Service Generated CSRs** set to **Yes**), then make sure that you set **Exportable** to **No**. |
| Private Key Trustee | Yes | (Optional) A Windows identity that is granted read access to the private key in the CAPI store.<br><br>When needed this is often the identity that the IIS Application Pool is running under. No special permissions are granted to the private key when left blank. |
| Delete Previous Cert and Key | Yes | Provides ability to delete the previous certificate and key, if necessary. |
| Pkcs#12 Encryption Algorithm | Yes | Choose the encryption algorithm used when exporting PKCS#12 files. Stronger algorithms are usually less compatible, especially with older software. |

## IIS Settings

| Field | Policy | Description |
|---|---|---|
| Update IIS | Yes | Controls whether Trust Protection Foundation will update IIS to use the installed certificate it just installed. |
| Web Site Name | Yes | The unique name of the IIS web site (Required if Update IIS is Yes). |
| Binding IP Address | No | The IP address for which the certificate will be bound to the IIS web site. If not specified the binding will be created for * which also appears as 'All Unassigned' in the Internet Information Services (IIS) Manager. |
| Binding Port | No | The port for which the certificate will be bound to the IIS web site (default is 443). |
| Binding Host Name | No | (Optional)Used to specify the TLS server name that IIS uses to determine which web site and certificate to present to server name indication (SNI)-enabled web clients.<br><br>This field applies only when you are provisioning to IIS 8.0, IIS 8.5, and IIS 10. This is because SNI (which allows binding SSL to host names rather than IP addresses) was added with the release of IIS 8.0. If Trust Protection Foundation attempts to provision to a target device running an earlier version of IIS, and you have specified a Binding Host Name value, this setting is simply ignored. |
| Create Binding | Yes | Controls whether Trust Protection Foundation creates a binding for the web site if one does not already exist. |

9. When you are finished, click **Save**.

For additional HSM configuration information, see "Managing applications using HSM-protected keys and Advanced Key Protect" on page 271 and "Configuring HSM-based remote key generation" on page 273.

### What's next?

After you've created an application object, here are other things you can do to manage the new application:

- On the application's **Settings** sub-tab:
  - Click  to push a certificate to its associated application.

    For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.
  - Click  **Reset** to stop processing the application and reset the status and stage.
  - Click  to reattempt installation of the certificate to its associated application, .
  - Click  **Validate Now** to validate the applications associated certificate.

    Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

    For more information, see "About certificate and application validation" on page 270.
- On the application object's **Validation** tab, you can configure validation settings for the application object.
- On an object's **General** tab:
  - Click the **Log** sub-tab to view any events that are triggered by the template object.
  - Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

## Citrix NetScaler configuration

The CyberArk Trust Protection Foundation™ NetScaler driver supports Citrix® NetScaler devices. This section provides the information you need to manage certificates on NetScaler devices.

For information about version dependencies, see "Certificate installation (provisioning) driver support" on page 38.

This chapter contains the following topics:

## NetScaler Certificate Lifecycle

The following table outlines the stages of the certificate lifecycle for certificates installed on supported NetScaler devices and the management level associated with each stage.

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| Stages 0-700 are performed by the NetScaler Application driver only if remote key generation is enabled. If the private key and CSR are locally generated on the Trust Protection Foundation server, stages 0-700 are performed by the X509Certificate Application driver.<br><br>The private key and CSR are remotely generated on the certificate's consumer application(s) if the **Generate Key/CSR on Application** option is enabled in the Certificate object. | | | | |
| 0 | StartProcessing | No processing. | | |
| 100 | CheckStore | No processing. | | |
| 200 | CreateConfigureStore | No processing. | | |
| 300 | CreateKey | Trust Protection Foundation creates the private key. | x | x |
| 400 | CreateCSR | Trust Protection Foundation creates the Certificate Signing Request (CSR). | x | x |
| 500 | PostCSR | Trust Protection Foundation submits the CSR to the Certificate Authority (CA).<br><br>If you post a manual CSR, this is the first stage of the certificate lifecycle. | x | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 600 | ApproveRequest | Trust Protection Foundation approves the certificate renewal at the CA. | x | x |
| 700 | RetrieveCertificate | Trust Protection Foundation retrieves the certificate from the CA. | x | x |
| 800 | InstallPrivateKey | Trust Protection Foundation installs the private key if the private key was not created on the host. | | x |
| 801 | InstallCertificate | Trust Protection Foundation installs the certificate on the NetScaler device. If the private key was not created on the device, Trust Protection Foundation installs the private key at this stage.<br><br>If **Include Root Chain** is selected, Trust Protection Foundation installs the root and the intermediate certificates on the NetScaler device. For more information, see "Install Certificate Chain" on page 407.<br><br>Root certificates can be discovered and brought under management, or they can be manually imported. For more information, see “Managing Root Certificates” on page 1. | | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| | | If **Import Only** is selected, Trust Protection Foundation does not provide the configuration required to make the certificate and private key functional on the device. For more information, see "Import Only" on page 408. | | |
| 802 | InstallCertificateChain | Trust Protection Foundation installs the root certificate chain file if it is available.<br><br>If there is an existing chain file, Trust Protection Foundation will install the new certificate chain file only if the **Overwrite Existing Chain** option is selected in the NetScaler Application object.<br><br>Root certificates can be discovered and brought under management, or they can be manually imported. For more information, see "Managing Root Certificates". | | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 803 | Associate Certificate and Private Key | Trust Protection Foundation associates the private key file with the certificate file in the application file system. | | x |
| 1400 | Revocation | Trust Protection Foundation submits a revocation request to the CA.<br>Certificate revocation is a certificate operation; it does not involve the application driver. | x | x |

## NetScaler Permission Requirements

When Trust Protection Foundation provisions certificates or conducts an onboard validation on a supported NetScaler device, the user account that Trust Protection Foundation uses to authenticate with the module must have the following permissions:

- Read and Write access to the directories where Trust Protection Foundation installs the certificate, private key, and root certificate chain files. These directories are defined in the NetScaler Application object. For more information, see "Creating a Citrix NetScaler application object" on page 404.
- Read and Write access to the Temp Directory defined in the Device object. For more information on the device object's configuration, see "Creating a device object in the Policy Tree" on page 239.

Make sure that the account used by Trust Protection Foundation to interact with NetScaler has least privilege (or greater) access.

Least privilege is achieved by creating a regular user on NetScaler and assigning to them a Command Policy with the following Command Spec:

```
(^stat\s+(cr|cs|lb|system|vpn|authentication))|(^(add|rm|show)\s+system\s+file\s+.*)|(^\S+\s+ssl\s+.*)|(^
(show|stat|sync)\s+HA\s+.*)|(^save\s+ns\s+config)|(^(switch|show)\s+ns\s+partition.*)
```

When provisioning to a partitioned NetScaler device make sure that the account used by Trust Protection Foundation to interact with NetScaler has permissions (for example, NetScaler SuperUser permission) to switch to the desired partition.

When provisioning to a partitioned NetScaler device with high availability (HA) configurations make sure that the account used by Trust Protection Foundation to interact with NetScaler has permissions to switch to the default partition.

## NetScaler prerequisite configuration

To enable Trust Protection Foundation to provision certificates on supported NetScaler devices, you must complete some prerequisite tasks related to the HTTPS connection method:

1. Verify that the Nitro API is enabled.

   Typically, the Nitro API is enabled by default on the versions supported by CyberArk (11.1 to 13.0)[1].

2. Make sure that all instances of Trust Protection Foundation that are interacting with NetScaler using Onboard Discovery, Provisioning, and/or File Validation have direct network access to it on port 443/tcp.

## Considerations for high availability with FIPS

When Trust Protection Foundation provisions a private key to a FIPS-enabled NetScaler appliance, the private key is stored in a local hardware security module (HSM). Although NetScaler synchronizes most of its settings when operating in a high availability configuration, it does not synchronize the FIPS key material stored in an HSM. Therefore, NetScaler contains references to the private keys but does not have access to the actual key material.

In order to copy FIPS keys between high availability nodes, you must configure Secure Information Management (SIM).

The only required configuration on Trust Protection Foundation is that when you are configuring the CyberArk NetScaler driver, you need to set **Use FIPS** to **Yes**. For more information, see "Use FIPS" on page 407 in the topic "Creating a Citrix NetScaler application object" on the next page.

[1]For specific supported versions and builds, please see System requirements for CyberArk components.

## Setting up SIM on a FIPS-enabled NetScaler device

**IMPORTANT** Because FIPS is not supported for ACMEv2, you cannot have FIPS enabled on any server that is used for ACMEv2.

To set up SIM on your FIPS-enabled NetScaler device, you must initialize and enable the SIM. If you do not complete SIM configuration, then provisioning will fail because the secondary node will contain a reference to a key that doesn't yet exist in the HSM.

To set up SIM, refer to your Citrix documentation: https://support.citrix.com/article/CTX118684.

**DID YOU KNOW?** To better understand what the CyberArk NetScaler driver needs to do to function in this scenario, refer to the following Citrix documentation, which outlines the tasks required to create and transfer FIPS keys: https://docs.citrix.com/en-us/netscaler/10-1/ns-tmg-wrapper-10-con/ns-ssl-wrapper-con-10/ns-tmg-fips-wrapper-con-10/ns-tmg-fips-create-transfer-fipskeys-con.html

## Creating a Citrix NetScaler application object

To enable Trust Protection Foundation to manage certificates installed on NetScaler devices, you must configure the NetScaler application object. This object provides the information Trust Protection Foundation needs to monitor, enroll, or provision certificates on its associated NetScaler devices.

**BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

**DID YOU KNOW?** When you add an installation to a certificate, you'll have the option of defining (and editing) this object during that process, which means that you don't have to log in to Policy Tree as the following procedure describes. And because the settings are the same, you can use this topic for information about each setting.

For more information, see Creating a certificate installation in Aperture in the *Certificate Management Guide*.

### To create and configure a Citrix NetScaler application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **NetScaler**.

3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.

4. (Optional) In the **Device Certificate** box, click ... to select and associate a certificate with the new application.

   > **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

   For detailed information on associating a certificate with an application, see Associating a certificate with an application object.

5. Under **General**, do the following:

   a. In the **Application Name** field, type a name for the new application.

   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.

   c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

      Default system notifications are sent to the contact identities. The default contact is the master administrator.

      > **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
      >
      > Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Under **Application Information**, do the following:

a. Click  next to **Application Credential** to browse for the credential object that you want to use to authenticate with the application.

> **DID YOU KNOW?** Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential might be a user name or private key credential; some drivers–such as F5, which is not SSH-based–can only use the user name credential for authentication.

> **NOTE** The user account you select must have *Read* and *Write* access to the Temporary, Private Key, and Certificate directories.
>
> If you need help with this step, see your system administrator.
>
> For more information, see Working with system credentials in the *Administration Guide*.

7. Complete the remaining settings for the application object by referring to the following table:

| Field | Policy | Description |
|---|---|---|
| **Certificate and Private Key Settings** | | |
| Install Certificate Chain | | Installs the intermediate certificate chain when the certificate is installed on the NetScaler device. |
| Use FIPS | | Creates the certificate and private key in accordance with the Federal Information Processing Standard (FIPS). This setting ensures the certificate that is installed on the current NetScaler device meets the FIPS.<br>**IMPORTANT** The NetScaler device must have the FIPS module.<br>For more information about FIPS, see "Considerations for high availability with FIPS" on page 403. |
| Private Key Credential | | The credential required to access the private key file for certificate renewal.<br>**To select a private key password credential**<br>a. Click ... to open the **Credential Selector** dialog.<br>b. Select the credential required to access the private key file for certificate renewal, and then click **Select**.<br>For more information, see Managing system credentials in the *CyberArk Trust Protection Foundation Administration Guide*.<br>**NOTE** Trust Protection Foundation does not include the private key password on the command line when performing key management operations. Instead, it provides the password when prompted. |

| Field | Policy | Description |
|---|---|---|
| Import Only |  | Designates that the certificate, private key, and any associated root or intermediate root certificates are only copied to the file system on the NetScaler device. Trust Protection Foundation does not provide the configuration required to make the certificate and private key functional on the device.<br><br>This configuration is provided for load balancing and failover environments where the NetScaler device configuration is automatically replicated to cluster or failover devices. In these environments, Trust Protection Foundation only needs to transfer the certificate and private key files. |
| Subfolder Relative Path | | Specify the path to an existing subfolder where you want Trust Protection Foundation to provision the certificate and key for your NetScaler virtual server.<br><br>During provisioning, Trust Protection Foundation will provision the certificates and private key to /nsconfig/ssl/*SubfolderRelativePath*. |
| Certificate File | No | The path and filename where Trust Protection Foundation installs the certificate on the NetScaler device.<br><br>**IMPORTANT** This setting must match the NetScaler device's certificate file configuration. For more information, refer to your NetScaler documentation.<br><br>The user account that Trust Protection Foundation uses to authenticate to the NetScaler device must have Read and Write access to the Certificate Directory. |
| Private Key File | No | The path and filename where Trust Protection Foundation installs the private key on the NetScaler device.<br><br>This setting must match the NetScaler device's private key file configuration. For more information, refer to your NetScaler documentation. |

| Field | Policy | Description |
|---|---|---|
| | | The user account that Trust Protection Foundation uses to authenticate to the NetScaler device must have Read and Write access to the Private Key Directory. |
| **SSL Configuration** | | |
| | | **NOTE** Be sure that you've completed prerequisite tasks before configuring SSL options. See "NetScaler prerequisite configuration" on page 403. |
| | | **TIP** If you enabled **Import Only** under **Certificate and Private Key Settings**, you can't modify **SSL Configuration** settings. |
| Bind Certificate To  | | Specify where Trust Protection Foundation should install the certificate.<br>From the **Bind Certificate To** list, do one of the following:<br>◦ Select **Virtual Server**, and then in **Virtual Server Name**, type the name of the NetScaler virtual server. This includes NetScaler Gateway Virtual Servers (beginning with Trust Protection Foundation version 20.3) and NetScaler Authentication Virtual Servers (beginning with Trust Protection Foundation version 24.1), where Trust Protection Foundation should install the certificate.<br>(Optional) Select the **SNI Certificate** check box if you are using a single virtual server to host multiple applications.<br>**TIP** Server name indication (SNI) allows the same virtual server to host multiple applications because it can present a unique certificate for each one. When SNI is not enabled, the virtual server can only have one certificate bound to it and therefore host only one application. For more information, see Server name indication (SNI) support for validation in the *Certificate Management Guide*.<br>◦ Select **Service**, and then in **Service Name**, type the name of |

| Field | Policy | Description |
|---|---|---|
| | | the NetScaler service.<br>◦ Select **Service Group**, and then in **Service Group Name**, type the name of the NetScaler service group.<br>◦ (Optional) If provisioning to an admin partition, specify it in the **Partition** field. If left blank, the driver provisions to the default partition. |

8. When you are finished, click **Save**.

## What's next?

After you've created an application object, here are other things you can do to manage the new application:

- On the application's **Settings** sub-tab:
  - Click  to push a certificate to its associated application.

    For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.
  - Click  **Reset** to stop processing the application and reset the status and stage.
  - Click  to reattempt installation of the certificate to its associated application, .
  - Click  **Validate Now** to validate the applications associated certificate.

    Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

    For more information, see "About certificate and application validation" on page 270.
- On the application object's **Validation** tab, you can configure validation settings for the application object.
- On an object's **General** tab:

- Click the **Log** sub-tab to view any events that are triggered by the template object.
- Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

## F5 LTM Advanced configuration–overview

You can use the F5 Local Traffic Manager (F5 LTM Advanced) driver to install and manage SSL certificates on your F5 LTM Advanced iControl platforms.

The F5 LTM Advanced driver supports SSL certificate provisioning, association with client and server profiles, and the administrative certificate on F5 LTM Advanced appliances and virtual editions through the iControl application programming interface (API).

**DID YOU KNOW?** Using *elliptic curve cryptography* (ECC) as your method of encrypting keys can help you prevent system outages. This is because the ECC algorithm is proving to be more secure than RSA.

For more information about ECC, see About RSA and elliptic curve cryptography (ECC) key algorithms in the *Certificate Authority and Hosting Platform Integration Guide*.

**DID YOU KNOW?** The F5 LTM Advanced driver detects name collisions automatically. If the driver encounters a certificate on the target F5 system that uses the same file name as the certificate to be provisioned, it compares them. If they are identical, the driver uses the existing certificate (on the F5 device) and continues normally through the remaining provisioning stages.

In cases where the certificates do not match, the driver attempts to install the certificate you want to provision. In order for this to succeed, the **Overwrite Certificate and Key** setting on the F5 F5 LTM Advanced Advanced template must be set to **Yes**. For more details, see "Configuring an F5 LTM Advanced application object" on page 425.

For information about version dependencies, see "Certificate installation (provisioning) driver support" on page 38.

This chapter contains the following topics:

## iControl considerations and background

This topic provides additional information about iControl compatibility and related background information.

### iControl compatibility issues caused by F5

The following table shows the versions of iControl that have known compatibility issues caused by specific F5 bugs:

| iControl Version | Trust Protection Foundation Error | Related F5 Error | Additional Notes |
|---|---|---|---|
| **11.0.0 through 11.2.0** | N/A | F5 Bug 364825:<br><br>SSL private keys that are stored in the FIPS module are not synchronized during ConfigSync operations.<br><br>https://support.f5.com/kb/en-us/solutions/public/13000/900/sol13929.html | This has been fixed in 11.2.1. |
| **11.5.4 HF2** | Error - 0107149e:3: Virtual server *virtual_server_name* has more than one clientssl/serverssl profile with same server name. | F5 Bug 614675:<br><br>iControl SOAP API call "LocalLB::ProfileClientSSL::create_v2" creates invalid profile | No GA hotfix as of November 2016. To work around this issue, create SSL profiles manually and then use Trust Protection Foundation only to update the certificate, key, and chain of existing SSL profiles. |

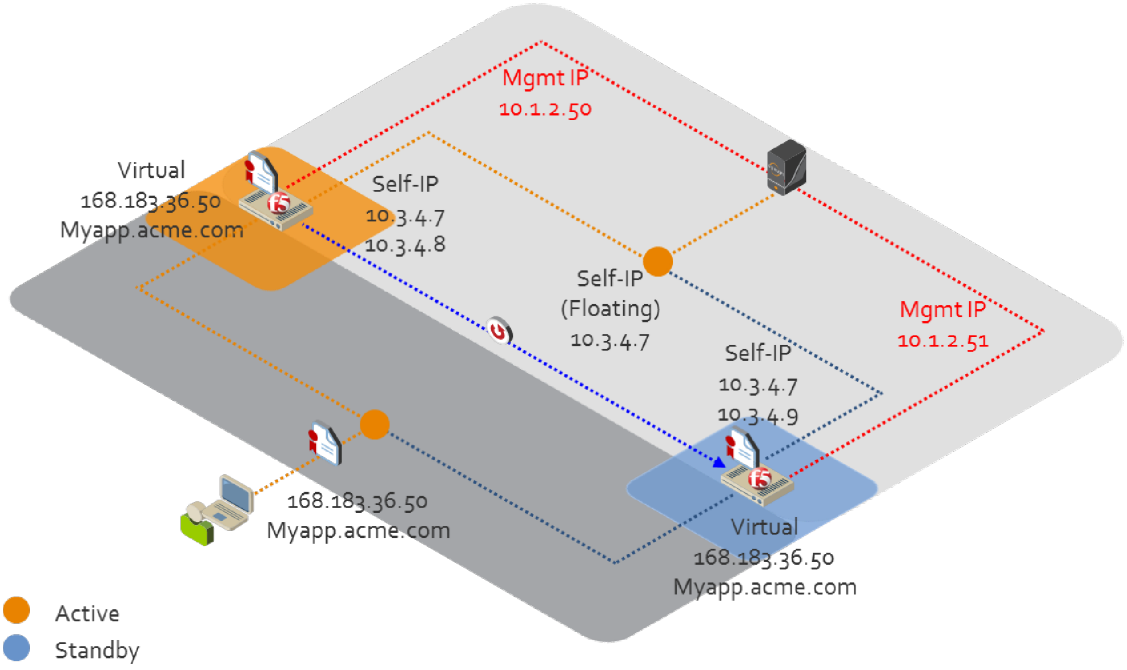
| iControl Version | Trust Protection Foundation Error | Related F5 Error | Additional Notes |
|---|---|---|---|
| 11.5.4 HF2 and 12.0.0 through 12.1.1 HF2 | Install Private Key (*private_key_name*) failed with error:<br><br>Exception caught in Management ::um:iControl:Management/KeyCertificate::key_import_from_pem()<br><br>01020066:3: The requested Certificate Key File (*private_key_file_name*) already exists in partition *partition_name*. | F5 Bug 614865:<br><br>The BIG-IP system may ignore iControl calls to overwrite an existing key or certificate<br><br>https://support.f5.com/kb/en-us/solutions/public/k/70/sol70340015.html | No GA hotfix as of November 2016. This issue causes minimal impact on Trust Protection Foundation 26.1 because the need to use the "overwrite" functionality has been minimized by comparing certificates when a name collision occurs.<br><br>https://support.f5.com/kb/en-us/products/big-ip_ltm/releasenotes/related/relnote-supplement-hotfix-bigip-12-1-0.html#A614865-1 |
| 12.0.0 through 12.0.0-HF2 | F5 Onboard Discovery out of memory error:<br><br>"Exception caught in Management::urn:iControl:Management/<br><br>KeyCertificate::certificate_export_to_pem()" | F5 Bug 564427:<br><br>iControl SOAP: memory leak in Management::KeyCertificate::get_certificate_list_v2<br><br>https://support.f5.com/kb/en-us/solutions/public/k/84/sol84349750.html | This has been fixed beginning with 12.0.0-HF3.<br><br>https://support.f5.com/kb/en-us/solutions/public/k/84/sol84349750.html |
| 12.0.0 | Install Certificate Chain (ca_bundle_name) failed with error: Unknown error 16908390 | F5 Bug 563760-1: | This has been fixed beginning with 12.1.0. |

| iControl Version | Trust Protection Foundation Error | Related F5 Error | Additional Notes |
|---|---|---|---|
| | | iControl call certificate_add_pem_to_bundle fails with the message that the certificate file already exists in the partition | https://support.f5.com/kb/en-us/products/big-ip_ltm/releasenotes/related/relnote-supplement-hotfix-bigip-12-0-0.html#A563760-1 |
| 13.1.0 through 13.1.0.8<br>13.1.1.0 through 13.1.1.4<br>14.0.0 through 14.0.0.4<br>14.1.0 through 14.1.0.2 | The operation has timed out. | ■ F5 Bug 758781-1:<br>iControl SOAP get_certificate_list commands take a long time to complete when there are a large number of certificates.<br>■ F5 Bug 726176:<br>This issue could prevent the ability to use a floating IP address for provisioning, which is the recommended configuration. | Results in extremely slow provisioning performance or fatal timeout errors when the target F5 is hosting hundreds of certificates.<br>Timeout errors seem to require 2,000 or more certificates in the partition.<br>**NOTE** F5 has not yet made a fix generally available for either of these bugs; therefore, if this issue affects you, please contact F5 Support to obtain an engineering hot fix. |

The FIPS synchronization issue with versions 11.0.1 through 11.2.0 cannot be detected by Trust Protection Foundation. Once provisioned, everything appears to have been installed and synchronized correctly. However, an SSL session on the F5 node that should have been synchronized to will fail, as the private key was not actually installed or updated. This is an issue outside of the control of Trust Protection Foundation and it is not an issue in iControl versions 11.2.1 or higher.

## Floating self-IP support

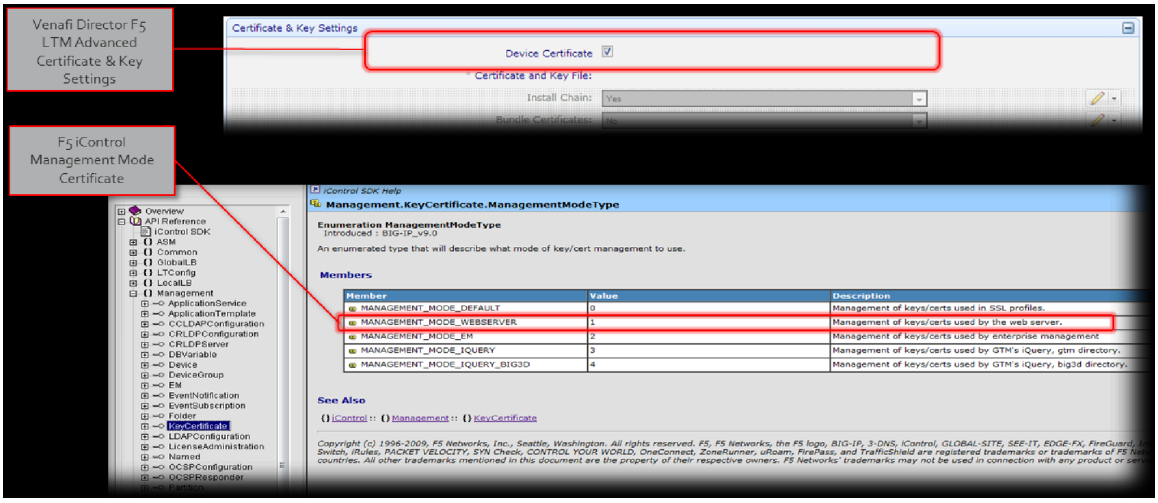
If you configure your F5 LTM Advanced in Trust Protection Foundation with the Floating Self-IP, it always provisions to the Active node which reduces the number of device objects you need to configure in Trust Protection Foundation. The following illustration shows an overview of how the Floating Self-IP works with a pair of F5s.




The F5 LTM Advanced driver will continually check the host where it is provisioning. This ensures that if a fail over or other change on the host occur that provision can be stopped and an error event log triggered.

## Provisioning the iControl Management Certificate

If you want to provision the iControl management session certificate, there is a configuration option in the F5 LTM Advanced driver. The following graphic shows the configuration option in the driver and the iControl console where the management certificate will be installed:




## Certificate file naming

It is necessary for the F5 LTM driver to manage the names of the certificate files in order to provide a workable method for certificate renewal, file replacement, disassociation of the old cert with the profile and association of the new certificate with the profile. This naming method follows similar rules as other provisioning drivers that control the file names. The F5 iControl API has a 63 character file name limitation. Automated file naming will be handled as follows:

Original common name:

- www.qa.departmentname.mydomain.net

Trust Protection Foundation-generated names:

- www.qa.departmentname.mydomain-1234567.crt
- www.qa.departmentname.mydomain-1234567.key

There are 55 characters available for the common name, 1 character for a dash and 7 characters for the first 7 digits of the certificate's serial number. The common name is truncated from the permission side, if necessary. If the serial number is less than 7 characters, it is prefixed (or padded) with zeros. For example, "SN: 1234" is given a prefix to become "-0001234").

## F5 LTM Advanced prerequisite configuration

To enable Trust Protection Foundation to provision certificates on BIG-IP F5 appliances, you must complete the following high-level tasks:

*For Trust Protection Foundation:*

1. Make sure that your version of F5 is compatible with the version of Trust Protection Foundation you're using.

   For compatibility information, see "Certificate installation (provisioning) driver support" on page 38.

2. Assign the Administrator role to the user account that Trust Protection Foundation uses to authenticate to the F5 appliance.

   For more information, see "F5 LTM Advanced permission requirements" on page 424.

3. Open access to the HTTPS port.

   Trust Protection Foundation uses the iControl protocol to manage certificates on BIG-IP F5 appliances. The iControl protocol uses the HTTPS port; therefore, Trust Protection Foundation must have access to the HTTPS port. The default HTTPS port is port 443.

*For F5:*

1. Using the BIG-IP web-based Configuration utility, create a pool that contains IP addresses of the content server nodes.

   For more information, refer to your F5 documentation.

2. On a F5 LTM Partition, create a virtual server that references the pool that is load balancing the SSL connections.

   For more information, refer to your F5 documentation.

3. (Conditional) If Basic Provisioning is being used, you must create the client and/or Server SSL Profiles necessary to support your application architecture. Client SSL Profiles define how F5 handles communications with clients such as web browsers. Server SSL Profiles define how the F5 interacts with content server nodes. Certificates are assigned to both SSL Profile types.

   For more information, refer to your F5 documentation.

4. In Trust Protection Foundation's Policy Tree, create a Device object for the machine where the F5 appliance resides.

   For more information, see "Creating a device object in the Policy Tree" on page 239.

5. In the Policy Tree, create and configure an application object for the F5 appliance.

   For more information on creating Application objects, see "Creating an application" on page 265.

6. In the Policy Tree, associate the F5 Application object with the certificates installed on the BIG-IP F5 appliance.

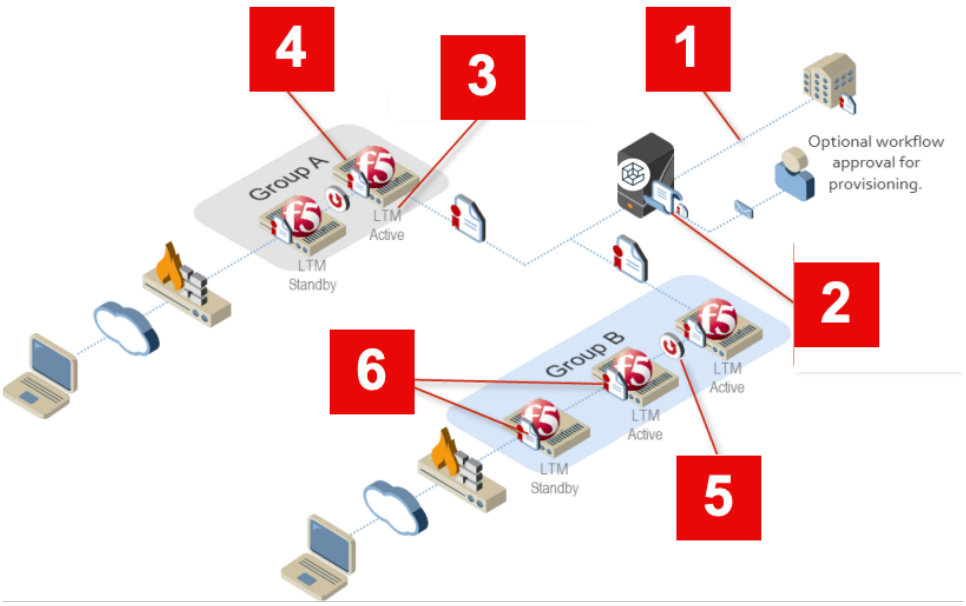
   For more information, see Associating a certificate with an application from the certificate object.

## What's Next?

To see the limitations of the F5 LTM Advanced driver and Trust Protection Foundation, see "Configuring an F5 LTM Advanced application object" on page 425.

## F5 LTM Advanced driver overview

For an overview of the F5 LTM Advanced driver's provisioning process, see the following diagram:




(see below for step summaries)

1. Certificate Manager - Self-Hosted enrolls a certificate from an internal or external CA.
2. Certificate Manager - Self-Hosted Admin associates the certificate with an F5 LTM application.
3. Provisioning checks F5 state (Active or Standby) and provisions to the configured state preference.
4. Archive existing certificate and associate the SSL Profile and Virtual Server.

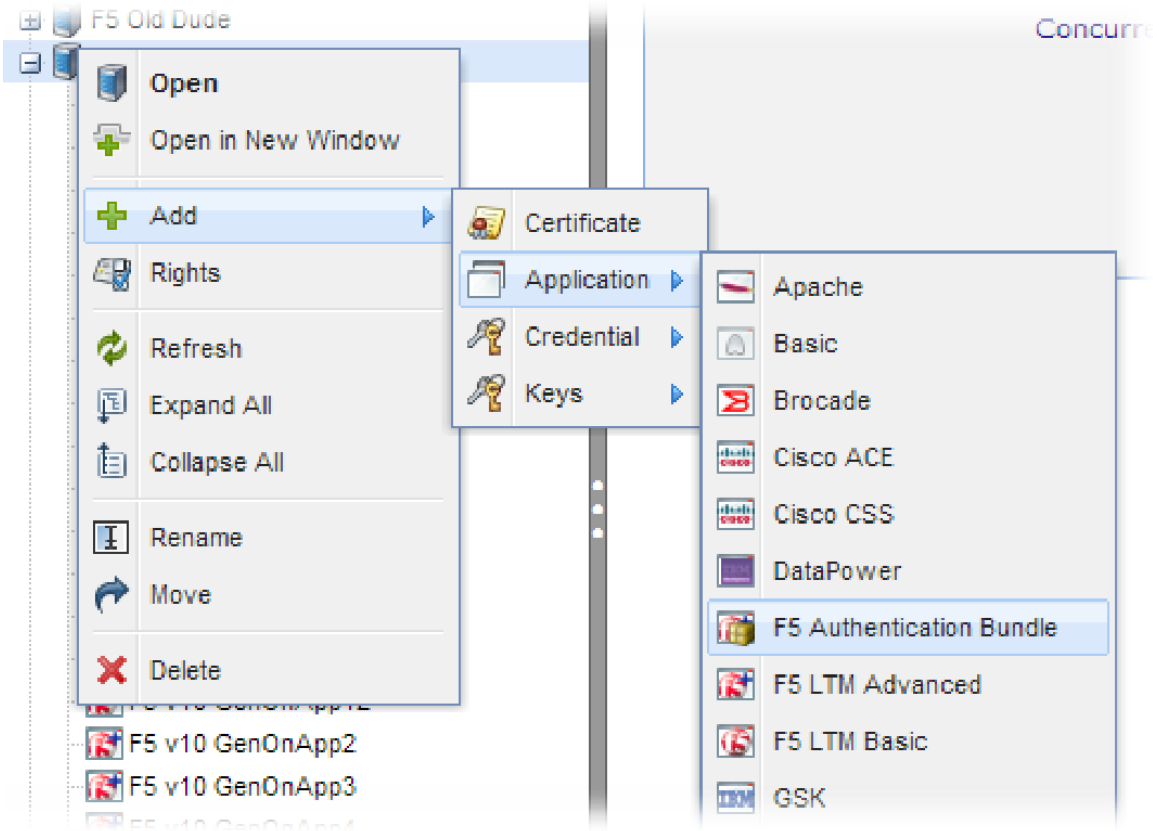
5. Execute ConfigSync if configured

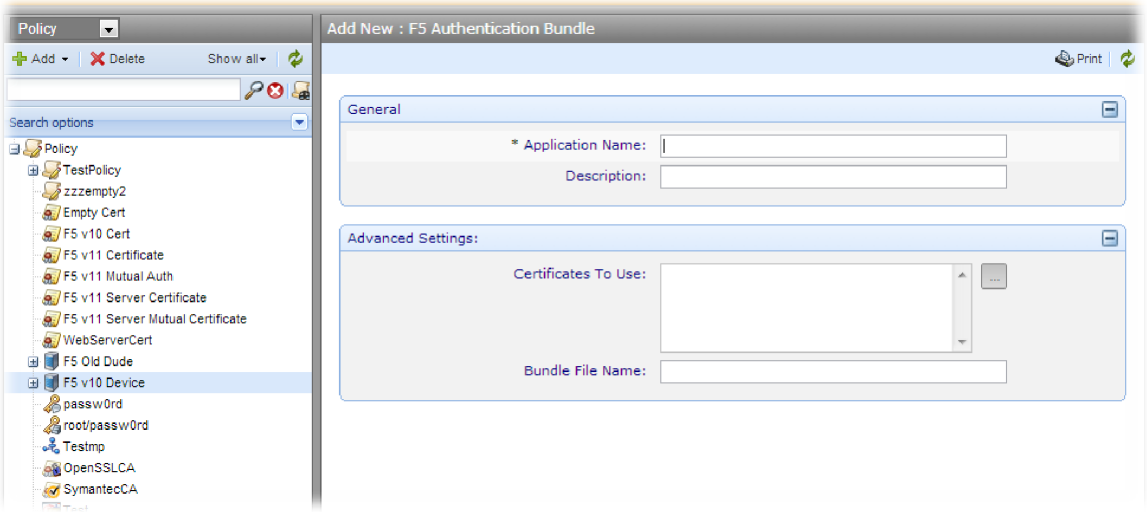
6. ConfigSync distributes certificates and settings to other nodes.

**NOTE** For information about the trust store, see About creating and configuring trust stores in the *CyberArk Trust Protection Foundation Certificate Management Guide*.

## About managing an F5 authentication bundle

From the Policy tree, you can right-click a device object and then choose **Add** > **Application** > **F5 Authentication Bundle**.






Enter the name of the bundle in the Application Name field (not be confused with the Application object name you'll be associating the bundle with). This is the name that will appear in the Policy Tree and in selection lists. You can also enter a description if there is any other information you wish to track for this object.

In the Advanced Settings section, click the ellipsis button to select one or more certificates from either the Policy or Roots tree that you want to include in the bundle trust store. You can also define the Bundle File Name that will be used when provisioning the bundle to the F5 Big IP.

These objects are only used in the F5 LTM Advanced application driver configuration settings.

## F5 LTM Advanced certificate lifecycle

The following table outlines the stages of the certificate lifecycle for certificates installed on F5 Big IP LTM appliances and virtual editions and the management level associated with each stage.

> **NOTE:** Remote generation with RSA 3072 bit keys is not supported.

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| Stages 0-400 are performed by the F5 LTM Advanced Advanced provisioning driver only if remote key generation is enabled. If the private key and CSR are locally generated on the Trust Protection Foundation server, stages 0-600 are performed by the X509Certificate enrollment driver. Stages 800-804 are performed by the F5 LTM Advanced provisioning driver.<br><br>The private key and CSR are remotely generated on the certificate's consumer application(s) if the **Generate Key/CSR on Application** option is enabled in the Certificate object. | | | | |
| 0 | StartProcessing | No processing. | | |
| 100 | CheckStore | No processing. | | |
| 200 | CreateConfigureStore | No processing. | | |
| 300 | CreatePrivateKey | Trust Protection Foundation creates the private key. Only done on application if set to 'Generate Keypair On Application', and if specified, creates it using a Federal Information Processing Standard (FIPS) -compliant algorithm). | x | x |
| 400 | CreateCSR | Trust Protection Foundation creates the Certificate Signing Request (CSR). | x | x |
| 500 | PostCSR | Trust Protection Foundation submits the CSR to the Certificate Authority (CA). | x | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| | | If you post a manual CSR, this is the first stage of the certificate lifecycle. | | |
| 600 | ApproveRequest | Trust Protection Foundation approves the certificate renewal at the CA. | x | x |
| 700 | Retrieve Certificate | Trust Protection Foundation retrieves the certificate from the CA. | x | x |
| 800 | Install Certificate Chain | Trust Protection Foundation installs the CA Bundle file if it is available.<br><br>If there is an existing bundle file, Trust Protection Foundation will install the new bundle file only if the **Overwrite Bundle** option is selected in the F5 Application object.<br><br>If there is an existing bundle file and the **Overwrite Bundle** option is not selected we simply add to the bundle if it is missing part of the chain. | | x |
| 805 | Install Private Key | If the non-encrypted private key was not created on the host, Trust Protection Foundation installs the private key. Provisioning of password-encrypted private keys is not supported. | | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 810 | Install Certificate | Trust Protection Foundation installs the certificate on the BIG-IP F5 network appliance. | | x |
| 815 | Install Advanced Settings Bundle | Trust Protection Foundation installs the F5 Authentication Bundle into the F5 trust store if configured to do so. | | x |
| 820 | Move Key To FIPS | If the private key was not created on the host and the **Use FIPS** option is selected in the F5 Application object, Trust Protection Foundation moves the private key to FIPS standards. | | x |
| 825 | Associate SSL Profile | Trust Protection Foundation associates the certificate and private key with an SSL Server or Client profile (based on the driver's configuration) so the server knows which certificate to use with which private key. | | x |
| 830 | Associate Virtual Server | Trust Protection Foundation associates the certificate and key with the configured Virtual Server. | | x |
| 835 | Delete Previously Used Certificate | Trust Protection Foundation checks to see if the certificate is associated with any other profiles; if it's not then archives the certificates and attempts to delete it. | | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 840 | ConfigSync | Trust Protection Foundation executes config synchronization if configured to do so. | | x |
| 845 | Config Save | Trust Protection Foundation executes Config Save command to save the BIG-IP configuration. | | x |
| 1200 | EndProcessing | Trust Protection Foundation completes the certificate processing and, if configured, runs a Validation check on the certificate and application. | | x |
| 1400 | Revocation | Trust Protection Foundation submits a revocation request to the CA.<br><br>Certificate revocation is a certificate operation; it does not involve the application driver. | x | x |

## F5 LTM Advanced permission requirements

When Trust Protection Foundation provisions certificates or conducts an onboard validation on an F5 LTM Advanced application, you must correctly configure permissions depending on the options you select.

*F5 LTM Advanced Permissions*: At minimum, the user account used by Trust Protection Foundation to access your F5 device must have the Administrator role, which can be assigned to a local user or a Remote Role Group for a non-local user. This is because the Administrator role is the least privilege required for interfacing with the iControl REST API.

## Configuring an F5 LTM Advanced application object

To enable Trust Protection Foundation to manage PEM files installed on BIG-IP F5 LTM network appliances using the iControl API, you must configure the F5 LTM Advanced application object. This object provides the information Trust Protection Foundation needs to monitor, enroll, or provision PEM files on associated network appliances.

> **BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

To see a list of supported provisioning drivers, see "Certificate installation (provisioning) driver support" on page 38.

| | |
|---|---|
| Overwrite Certificate and Key | Assets that are in use cannot be overwritten and replaced. |
| Delete Previous Cert and Key | Does not use generational management. |
| Password-encrypted private keys | Onboard Discovery cannot import this type of key. |
| Certificate Partition | Profile and certificate can exist in different partitions; but this configuration is not supported for provisioning and will result in Onboard Validation failures. |
| Onboard Validation | Does not validate the virtual server and profile associations of the certificate |
| SNI | SNI (Server Name Indication) profiles are supported by Trust Protection Foundation and the F5 LTM Advanced driver.<br><br>**NOTE** You can use the CyberArk driver to introduce SNI configurations even if they do not yet exist on F5, provided that they do not affect existing F5 settings. Also, you cannot overwrite existing F5 settings. |

**DID YOU KNOW?** When you add an installation to a certificate, you'll have the option of defining (and editing) this object during that process, which means that you don't have to log in to Policy Tree as the following procedure describes. And because the settings are the same, you can use this topic for information about each setting.

For more information, see Creating a certificate installation in Aperture in the *Certificate Management Guide*.

### To create and configure an F5 LTM Advanced application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **F5LTM** .
3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.
4. (Optional) In the **Device Certificate** box, click ... to select and associate a certificate with the new application.

   **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

   For detailed information on associating a certificate with an application object, see Associating a certificate with an application object.
5. Under **General**, do the following:
   a. In the **Application Name** field, type a name for the new application.
   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.

c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

Default system notifications are sent to the contact identities. The default contact is the master administrator.

> **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
>
> Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Under **Application Information**, do the following:

a. Click  next to **Application Credential** to browse for the credential object that you want to use to authenticate with the application.

> **DID YOU KNOW?** Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential might be a user name or private key credential; some drivers–such as F5, which is not SSH-based–can only use the user name credential for authentication.

> **NOTE** The user account you select must have *Read* and *Write* access to the Temporary, Private Key, and Certificate directories.
>
> If you need help with this step, see your system administrator.
>
> For more information, see Working with system credentials in the *Administration Guide*.

> **DID YOU KNOW?** The **Connection Method** is the protocol that Trust Protection Foundation uses to connect to the server and manage the certificates installed on that server. In an application object's settings, this field is typically read-only.

b. (Read Only) The **iControl Version** field shows the last detected version of iControl when the driver was connected to the F5 host device/application.

c. Click the **Connection Method** list, click the protocol to use–HTTPS or SSH–and then in the **Port** field, specify the associated port number.

d. In the **HTTPS Port** field, type the port that Trust Protection Foundation should use to communicate with your appliance.

> **DID YOU KNOW?** By default, all communication between the F5 LTM Advanced driver and F5 (iControl) is done using port 443 except for *Workflow command injection*, which uses port 22 (SSH) by default. However, workflow command injection isn't commonly used with F5 for several reasons, including the following:
>
> - The F5 driver is robust enough that it typically does not require additional functionality
> - Would require elevated permissions
> - Would also require command line access

e. (Conditional) In the **SSH Port** field, specify the port number that Trust Protection Foundation should use to communicate with the appliance via an SSH connection.

The default SSH port assignment is 22.

7. Complete the F5 F5 LTM Advanced Advanced application settings by referring to the following table:

| Field | Policy | Description |
|---|---|---|
| **Certificate & Key Settings** | | The following are server-specific certificate settings. They are referenced only when you associate a certificate with subordinate BIG-IP F5 LTM Application objects. |
| Device Certificate | | Select this option if you wish to provision the F5 F5 LTM Advanced's iControl console management certificate. |
| Provisioning Mode |  | This setting overrides all provisioning settings except for *Certificate & Key Settings*. It provisions the PEM file that contains a certificate, private key, and (optional) chain bundle to the F5 Big IP LTM keystore. This setting requires the Certificate Manager - Self-Hosted user role on the LTM.<br><br>If you want encryption, be sure to configure the *Private Key Credential*. Otherwise, the private key is non-encrypted. Choose the Provisioning Mode:<br><br>◦ *Advanced*: This is the default and it not only provisions the certificate, private key, and (optionally) chain bundle, it binds those assets to an SSL profile and, if the SSL profile doesn't already exist, it creates the profile and assigns it to a virtual server<br><br>◦ *Basic with configurable Certificate Name*: This only provisions the certificate, private key, and (optionally) chain bundle and it allows for the user to specify their own name for the certificate. If an existing certificate is found with the requested name then a numeric suffix will be appended to the name until uniqueness is achieved.<br><br>◦ *Basic with automatic Certificate Name*: This only provisions the certificate, private key, and (optionally) chain bundle and uses the same automatic naming convention used by Advanced to ensure the certificate name is unique. |

| Field | Policy | Description |
|---|---|---|
| | | The naming convention is: *<Common Name>-<Expiration Date as DDMMMYY>-<Serial Number>*<br><br>If the Common Name results in a name that would exceed the character limit of the F5, characters are trimmed from the end of the common name until an allowable length is achieved. |
| Certificate & Key File | | This read-only field displays the filename of the certificate associated with the current F5 LTM Application object once it is provisioned. If it has not yet been provisioned there will be no value in this field. This setting applies when *Provisioning Mode* is Advanced or Basic. |
| Private Key Credential | | The password that encrypts the private key. |
| Force Profile Update | | Used to handle situations where the password has changed between generations of the certificate.<br><br>When set to No–which is the default setting–then Trust Protection Foundation displays an error stating that provisioning could not be done because the password has changed since the last time it was provisioned.<br><br>When set to Yes, then Trust Protection Foundation continues with provisioning; however, there will be a brief service interruption during provisioning. |
| Install Chain |  | Option to install the chain on the F5 LTM application. |
| Bundle Certificates |  | Option to bundle the root and intermediate root certificates with the certificate file installed on the current BIG-IP F5 LTM appliance. |
| Overwrite Chain File |  | Overwrites the existing certificate chain file when it provisions a certificate on the BIG-IP F5 LTM appliance.<br><br>If you do not select this option, Trust Protection Foundation cannot provision certificates if there is an existing certificate on this network appliance. |

| Field | Policy | Description |
|---|---|---|
| CA Chain File | | Filename of the chain file associated with the current BIG-IP F5 LTM Application object. |
| Key Security Type |  | Creates the certificate and private key in accordance with the Federal Information Processing Standard (FIPS) or stored on an external HSM (NetHSM). This setting ensures the certificate that is installed on the current BIG-IP F5 LTM appliance meets FIPS requirements or the private key is created and stored on an external HSM (the associated certificate must set "Generate Key/CSR on Application" to "Yes").<br><br>On the F5, FIPS and NetHSM require configuration. See the F5 documentation.<br><br>**NOTE** Advanced Key Protect must be Enabled to use the NetHSM (external HSM) option. |
| Overwrite Certificate and Key |  | Overwrites the existing certificate and key file when it provisions a certificate and non-encrypted private key on the BIG-IP F5 LTM appliance. |
| Delete Previous Cert and Key |  | This option allows you to choose if you want to attempt to delete the previous certificate and key if they exists and if they are not associated with another SSL profile on the F5 LTM application. |
| **High Availability Settings** | | The following settings let you configure options for F5 LTM applications running in high availability (HA) pairs or clusters.<br><br>**DID YOU KNOW?** These options provide a work-around to the more common (and often recommended) floating IP configuration for F5s running in HA mode. If using a Floating IP is not an option, then these HA options can help you to provision your F5s that are configured for failover. |

| Field | Policy | Description |
|---|---|---|
| Provisioning To |  | CyberArk recommends that you provision to the floating management IP of the F5 Big-IP, in which case you'll always be provisioning to an active node. However, if for some reason you can't (or choose not to) provision to the floating IP, then you must provision directly to each appliance. In this case, you'll need to decide if you want Trust Protection Foundation to update the Active node, Standby node, or both (Ignore Failover State).<br><br>You can select between *Standalone*, *Active*, *Standby* or *Ignore* modes.<br><br>◦ **Standalone**: use when you have only one server.<br><br>◦ **Active**: use when you want Trust Protection Foundation to provision only to active F5 appliances.<br><br>◦ **Standby**: use when you want Trust Protection Foundation to provision only to F5 appliances that are in stand by mode.<br><br>◦ **Ignore**: use when you want Trust Protection Foundation to provision to F5 appliances regardless of whether or not they're in active or stand-by modes.<br><br>When selected, the Ignore option bypasses logic in the provisioning steps where Trust Protection Foundation checks the state of the F5 Big IP node it's provisioning to where there is a pair or cluster. This meets a use-case where the F5 Big IP LTM pair or cluster is not configured with a floating IP address and you do not want the state to be checked or considered when provisioning the certificate to the defined host because it may change over time.<br><br>Keep the following in mind when considering provisioning directly to each F5 appliance:<br><br>◦ Provisioning directly requires twice the number of F5 application objects: one set for the first appliance and a |

| Field | Policy | Description |
|---|---|---|
| | | duplicate set for the second.<br>◦ If you choose Active or Standby, then provisioning to half of those F5 apps is going to fail because the appliance isn't in the HA state required for Trust Protection Foundation to provision to it; this also means that after provisioning is complete, you'll have to perform additional work to clear errors.<br>**TIP** You can automate the additional work using Adaptable Log Channel or Adaptable Workflow.<br>◦ If you choose Ignore Failover State, then both appliances are updated; however, about half of the work that the F5 driver does will be lost; this is because the Config Sync will overwrite the updates made to the target of the sync. |
| Config Sync |  | This option allows you to choose the if you would like the driver to execute a config synchronization after the certificate and key is installed on the F5 LTM application running in HA mode. |
| **SSL Profile Settings** | The following settings allow you to configure the SSL Profile name, type, parent profile, virtual server and partition with which the certificate will be associated on the F5 LTM application. | |
| SSL Profile | NA | Specify the name of the F5 LTM SSL Profile you wish to create (if it does not exists) or with which to associate the certificate. |
| SSL Profile Type | | Use to select the F5 LTM profile type: Server or Client. |
| Parent SSL Profile |  | Specify the name of the F5 LTM Parent SSL Profile that the SSL Profile will use to inherit its settings from. If no value is specified here, the F5 LTM Default SSL Profile is used. |

| Field | Policy | Description |
|---|---|---|
| SSL Partition |  | Specify the name of the F5 LTM Partition that the Certificate and SSL Profile will be installed in. If no value is specified, the default partition is used. |
| SNI Server Name | | Type the name of the SNI server to enable provisioning of certificates that are being used in SNI configurations.<br><br>This field is enabled when you specify either Client or Server as the SSL Profile Type. When provisioning, the SNI Server Name is automatically updated to match the settings of the actual SSL Profile on the F5 if the profile already exists. |
| SNI Default | | When set, this SSL profile is used as the default profile when there is no match for the server name, or when the client does not support or use SNI.<br><br>This setting is available for both Client and Server SSL profiles. |

| Field | Policy | Description |
|---|---|---|
| | | **DID YOU KNOW?** You can configure SNI Server Name and SNI Default for both Client and Server SSL profiles on F5 devices–not just client profiles. This means you can use SNI for both inbound and outbound SSL/TLS connections managed by your F5.<br><br>Suppose you have one F5 virtual server hosting multiple websites, all resolving to the same IP address. You want each site to present its own SSL certificate. Before SNI, you had to:<br><br>◦ Run each site on a different port (e.g., https://site1.net:443, https://site2.net:8080), which is inconvenient for users.<br><br>◦ Assign a dedicated IP address to each site, which is inefficient and costly.<br><br>◦ Use a single certificate for all sites, which leads to browser warnings for domain mismatches.<br><br>SNI (Server Name Indication) solves these problems by allowing the F5 to present the correct certificate based on the hostname requested by the client–even when multiple SSL profiles are associated with a single virtual server. With support for both client and server SSL profiles:<br><br>◦ You can specify the SNI Server Name for any SSL profile type, ensuring the right certificate is presented for each hostname.<br><br>◦ The SNI Default option lets you designate a fallback profile for connections that don't specify a hostname or don't support SNI.<br><br>This expanded support gives you greater flexibility and security for complex, multi-site environments–whether you're managing inbound client connections or outbound server connections. |

| Field | Policy | Description |
|---|---|---|
| **SSL Profile Association** | | The following settings require information to associate a server SSL profile to an F5 monitor or an existing virtual server. |
| Associate SSL Profile To | | You can select between *Monitor*, *Virtual Server*, or *No Association*.<br>◦ **Monitor**: Use to associate an SSL profile to an F5 monitor. Associating monitor to SSL Profile will work only with BIG IP versions greater than 13.1.0.<br>The F5 monitor type must be https or http2.<br>http2 monitors supported are from versions V15.1 or greater.<br>◦ **Virtual Server**: Use when you have an existing virtual server that is running on an F5 LTM. Always set up your virtual server prior to creating this F5 LTM application.<br>◦ **No Association**: Use to provision without associating to any virtual server or monitor. |
| Virtual Server |  | This field only applies when Virtual Server is selected to associated the SSL profile to. Specify the name of an existing F5 LTM Virtual Server that the SSL Profile will be associated with. This is a required field. |
| Virtual Server Partition |  | This field only applies when Virtual server is selected to associated the SSL profile to. Specify the name of an existing F5 LTM Partition that runs the virtual server. If no value is specified the default partition will be used. |
| Monitor | | This field only applies when Monitor is selected to associated the SSL profile to. Specify the name of the monitor that the SSL profile with be associated with . This is a required field. |
| Monitor Partition | | This field only applies when Monitor is selected to associated the SSL profile to. (Optional) Specify the monitor partition. If no value is specified, the common partition will be used. |

| Field | Policy | Description |
|---|---|---|
| **Advanced Settings** | | The following settings allow you to configure advanced settings for the SSL profile related to Mutual Authentication and truststore bundle provisioning. |
| Use Advanced Settings |  | This field and all of the others (except the last two display fields) can be controlled with policy. This checkbox turns on or off all the fields in the Advanced Settings section. If you are not using Mutual Authentication on your F5 LTMs, it can help the driver to be less complicated for your Trust Protection Foundation users who are setting up F5 Big IP LTM application. |
| Client Certificate |  | This field is only available if you choose 'Client' in the SSL Profile Type field in the SSL section in the F5 LTM Advanced driver settings. For a Client SSL Profile type the choices are "ignore", "require", "request" or "auto". This setting specifies the way the system handles client certificates. The default is Ignore.<br><br>**Ignore**: Specifies that the system ignores certificates from client systems. Server & Client Profiles<br><br>**Require**: Specifies that the system requires a client to present a valid certificate. Server & Client Profile<br><br>**Auto**: Specifies that the system ignores a client certificate until an authentication module requests one. Client Only<br><br>**Request**: Specifies that the system requests a valid certificate from a client but always authenticate the client. Client Only |
| Server Certificate |  | This field is only available if you choose 'Server' in the SSL Profile Type field in the SSL section in the F5 LTM Advanced driver settings. For a Server SSL Profile type, the choices in this drop-down are "ignore" or "require",<br><br>**Ignore**: Specifies that the system ignores certificates from client systems. Server & Client Profiles<br><br>**Require**: Specifies that the system requires a client to present a valid certificate. Server & Client Profile |

| Field | Policy | Description |
|---|---|---|
| Frequency |  | For both Server and Client SSL Profiles, the choices here are "Once" or "Always". This value defines the frequency of client authentication for an SSL session. The default is Once, which specifies that the system authenticates the client once for an SSL session. Always specifies that the system authenticates the client once for an SSL session and also upon reuse of that session. |
| Chain Traversal Depth |  | You can choose a number from 1 to 9, with 9 being the default depth value. Specifies the maximum number of certificates to be traversed in a client certificate chain. |
| Certificate Bundle |  | This field lets you select an *F5 Authentication bundle* for one or more certificates that will be provisioned to the F5 and associated with the profile as the authentication bundle used for mutual authentication.<br>Click  to make your selection from the Policy tree.<br>If there are no bundles listed, you must first create one. For more information about creating an authentication bundle, see "About managing an F5 authentication bundle" on page 419 in the *CyberArk Trust Protection Foundation Certificate Management Guide*. |

| Field | Policy | Description |
|---|---|---|
| Authentication Name |  | This field only applies to Server SSL Profiles and will not be selectable when a Client SSL Profile type if selected. This field specifies a Common Name (CN) that is embedded in a server certificate. The system authenticates a server based on the specified CN. The CN name must be a valid fully qualified domain name (FQDN), and hence the authenticate name specified must also be a FQDN. Note that the Domain Name System (DNS) does not allow the underscore ( _) character in CN names, although it does allow the dash ( -) character.<br><br>**IMPORTANT** If you select **Require in Server Certificate**, make sure to specify a value in **Authenticate Name** as well. A blank Authenticate Name field means that everyone is authenticated, even though you have specified Require as the Server Certificate setting. |
| Trusted CA File | | This field is for display purposes only, as Trust Protection Foundation controls the name of the certificate file and truststore bundle file. For a Server SSL Profile this could be a single self-signed certificate or one or more certificates in a truststore bundle. |
| Advertised CA File | | This field is for display purposes only and only applies to Server SSL Profile types. |

8. When you are finished, click **Save**.

### What's next?

After you've created an application object, here are other things you can do to manage the new application:

- On the application's **Settings** sub-tab:
  - Click  to push a certificate to its associated application.

    For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.
  - Click  **Reset** to stop processing the application and reset the status and stage.

- Click  to reattempt installation of the certificate to its associated application, .
- Click  **Validate Now** to validate the applications associated certificate.

  Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

  For more information, see "About certificate and application validation" on page 270.

- On the application object's **Validation** tab, you can configure validation settings for the application object.
- On an object's **General** tab:
  - Click the **Log** sub-tab to view any events that are triggered by the template object.
  - Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

# Integrating Google Cloud Load Balancer with CyberArk

The CyberArk Trust Protection Foundation™ Google Cloud Load Balancer driver supports global and regional (HTTPS) proxies, which are used by external and internal HTTPS load balancers.

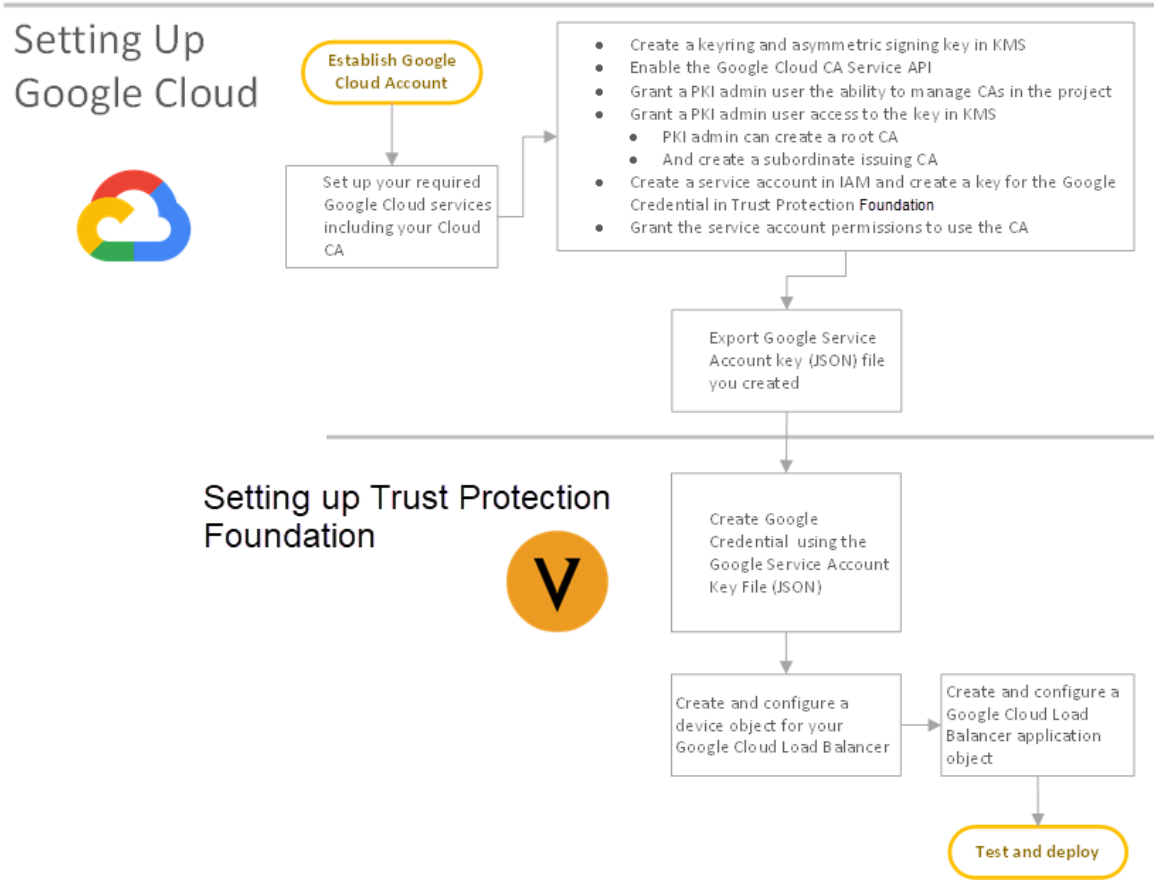
This topic provides the information you need to manage certificates on your external load balancers.

For information about version dependencies, see "Certificate installation (provisioning) driver support" on page 38.

> **DID YOU KNOW?** Keep in mind that certificates you plan to provision must have the encryption algorithm set to either RSA-2048 or ECDSA P-256. Click here for more details for more details found in the Google Cloud Load Balancer documentation.

## First things first

Before you start configuring integrations with Google Cloud Load Balancer, carefully review this *high-level* diagram. It can really help you complete your integration faster.




**TIP** For a rather nifty and helpful overview of the steps for integrating applications with CyberArk, look here.

To complete your integration, complete the following steps:

- "Step 1: Setting up Google Cloud" below
- "Step 2: Setting up CyberArk for integration with Google Cloud Load Balancer" on the next page

## Step 1: Setting up Google Cloud

Google knows its stuff best, so we won't presume to give you all the details when using Google products. We strongly suggest that you carefully review and follow Google documentation to establish your Google Cloud account.

**IMPORTANT** Here are the minimum permissions you must set up on the Google Cloud side:

- compute.sslCertificates.create
- compute.sslCertificates.delete
- compute.sslCertificates.get
- compute.targetHttpsProxies.get
- compute.targetHttpsProxies.setSslCertificates
- compute.targetSslProxies.get
- compute.targetSslProxies.setSslCertificates
- compute.regionSslCertificates.create
- compute.regionSslCertificates.delete
- compute.regionSslCertificates.get
- compute.regionTargetHttpsProxies.get
- compute.regionTargetHttpsProxies.setSslCertificates
- compute.regions.list

Not required but additional permissions used by Google Cloud Load Balancer application driver:

- compute.targetHttpsProxies.create
- compute.regionTargetHttpsProxies.create
- compute.regionUrlMaps.use
- compute.urlMaps.use

When you've done the Google thing, then you're ready to configure the CyberArk side.

### Step 2: Setting up CyberArk for integration with Google Cloud Load Balancer

Make sure you've already set up your Google Cloud account and configured your Load Balancer. Your Google account must be up and running for CyberArk to connect and do its thing.

## First things first

Let's take care of prerequisite steps first.

- Make sure you have the right permissions to the Policy folder in WebAdmin where you'll be doing the configuration. Specifically, you'll need
  - View/Read/Write permissions for uploading your Google Service Account Key File
- To create the Google Cloud Load Balancer application object, you need View, Read, Write, and Create permissions on the corresponding policy (folder) where you create the object.
- Read and Write access to the Temp Directory defined in the Device object.
- Make sure you have your Google Service Account Key File ready for uploading to CyberArk. You'll need it!

> **TIP** Not sure how to set permissions on Policy folders? Look here, and then come back when you're ready.

## Creating a Google Credential

A Google credential object is used by CyberArk Trust Protection Foundation to store your Google Cloud Service Account Key file. Your Google Service Account is used by CyberArk Trust Protection Foundation to access your Google Cloud CA and Google Cloud Load Balancer.

> **TIP** If you're a developer, you might be interested in knowing that the Google credential is defined in the schema using the Google Credential class and implemented in Venafi.Drivers.GoogleCredential: CredentialObject.

### To create a new Google Credential

1. From the Certificate Manager - Self-Hosted menu bar, click **Policy Tree**.
2. From the **Policy** tree, select the folder where you want to create the Google Credential, and then click **Add** > **Credential** > **Google Credential**.
3. In **Credential Name**, type a name for the new credential.
4. Click **Upload Google Service Account Key File** (button near the top of the page), locate the JSON

file, and then click **Upload**.

5. Click **Save**.

## Creating a device object for Google Cloud Load Balancer

There's actually nothing special or unusual about creating a device object for Google Cloud Load Balancer. But it's good to note that for this driver, device details (host/login credentials) *are irrelevant*. That's because the Google Credential takes care of that for you. So no need to fill in those details when you create the device object.

If you've never created a device object on the CyberArk Platform, these steps can help you through it.

Either way, come back here when you're finished and we'll take the next (and final) step.

## Creating a Google Cloud Load Balancer application object

Trust Protection Foundation uses the information you'll supply in a Google Cloud Load Balancer application object to manage certificates installed on Google Cloud Load Balancer devices.

### To create and configure a Google Cloud Load Balancer application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **Google Cloud Load Balancer**.
3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.
4. (Optional) In the **Device Certificate** box, click ... to select and associate a certificate with the new application.

   > **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

For detailed information on associating a certificate with an application, see Associating a certificate with an application object.

5. Under **General**, do the following:

   a. In the **Application Name** field, type a name for the new application.

   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.

   c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

      Default system notifications are sent to the contact identities. The default contact is the master administrator.

      > **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
      >
      > Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

   d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

      The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

   e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

      You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Under **Application Information**, click  to select your Google Credential that you created earlier.

7. The **Project ID** is populated using the project from the valid Google Credential. This provides a way to specify in which project the Private CA is placed. The value of the field is extracted from the Google Credential, which the user has the option to edit.

8. Under **Installation Settings**, specify target proxy server information.

   If you specify **Internal HTTPS**, as the **Target Proxy Type**, make sure to specify its correct region in the **Target Region** field.

   You can provision to an existing proxy; or a new one can be created if a correct and existing Target Resource is specified.

   For HTTPS proxies, type the existing URL Map in **Target Resource**; and for SSL proxies, enter an existing Backend Service.

   The provisioned certificate will not bind to a Google resource if the Target Proxy Name is left empty. Otherwise, we will bind the certificate to the Target Proxy Name.

   > **DID YOU KNOW?** If you try to provision to a Google Cloud Load Balancer and you either 1) left the Target Resource field blank, or 2) had specified a target resource that does not exist, then Trust Protection Foundation attempts to retrieve the target resource from the cloud service automatically and populates the Target Resource field with the relevant one. Either way, a warning is logged for the Google Cloud Load Balancer application object.

9. When you're finished, click **Save**.

## HashiCorp Vault PKI–overview

The HashiCorp Vault PKI application is the representation of a Vault PKI secrets engine in Trust Protection Foundation. Once you set up the HashiCorp Vault and application, the Vault automatically issues end entity certificates to users

The driver:

- Creates and enforces the role policy that the CA requires to issue or sign end-entity certificates. Only policy folders created by the Web SDK Role method can be provisioned to the Vault.
- Automates the lifecycle of subordinate CA certificates in a HashiCorp PKI secrets engine.

Unlike most drivers, you set up and manage certificate issuance with Web SDK methods. There are minimal UI elements. Instead, you use Role and CA methods from the Web SDK.

After calling an API method, you can view the result in the Policy tree and Vault. Additional information is available on the application object **Support** and **Log** tabs for troubleshooting.

## HashiCorp Vault permission requirements

In HashiCorp, be sure you use a policy to grant access to Vault PKI paths. To create the policy, you can use the HashiCorp Configuration Language (HCL). For more information, see https://www.vaultproject.io/docs/concepts/policies.

The HashiCorp Vault PKI application driver requires permissions to the *pki* path of a mounted PKI secrets engine:

| HashiCorpPath | Required Permissions |
|---|---|
| *pki/** | Read, and list permissions |
| *pki/config/** | Read, list, create, and update permissions to allow the driver to set the CRL and OCSP addresses for the CA. |
| *pki/intermediate/** | Read, list, create, and update permissions to allow the driver to generate a key pair and CSR for the CA, and install the subordinate CA certificate. |
| *pki/roles/** | Read, list, create, and update permissions to allow the driver to create and update roles for the PKI secrets engine. |

## HashiCorp Vault PKI roles

PKI roles authorize HashiCorp users who can request user certificates. Role information appears in both the HashiCorp Vault PKI application and the HashiCorp Vault PKI secrets engine.

Via the Web SDK, the HashiCorp Vault PKI driver:

- Helps with the creation of roles in the PKI secrets engine.
- Ensures that roles comply with the enterprise security policy for certificates.
- Performs role operations during daily or on demand CA certificate validation. A greater level of awareness than a standard certificate validation is required. If an error occurs with one of the role operations, the certificate goes into an error state.

You can use Web SDK methods to manage HashiCorp roles. For example, POST PKI/HashiCorp/Role in the *Developer's Guide* creates a role. The role, is represented by a standard Policy folder object. The role policy constrains the properties of the end-entity certificates that can be issued by the Vault PKI secrets engine.

After creation in Trust Protection Foundation, the role is ready to provision to the Vault. For more information, see " Creating and managing the HashiCorp PKI Vault" on page 451.

**WARNING!** Do not manually create role policy folders for HashiCorp Vault PKI. Instead use, POST PKI/HashiCorp/Role. This method allows for creation and assignment of special attributes that the driver cannot create through WebAdmin or Aperture.

## HashiCorp Vault PKI application creation

When POST PKI/HashiCorp/CA creates the HashiCorp Vault PKI application, it is represented by these objects:

- **Device**: The network location of the Vault server and the credentials that Trust Protection Foundation uses when interacting with HashiCorp.
- **HashiCorp Vault PKI application**: The management configuration for the secrets engine running in the Vault server. The name of the application object matches the name of the secrets engine.
- **Certificate**: The enrollment/renewal settings of the subordinate CA certificate for the secrets engine indicated by the application.

## HashiCorp Vault PKI provisioning

During provisioning of a subordinate CA certificate to a HashiCorp Vault PKI secrets engine, the driver:

- Invokes the Vault API to generate the key pair and CSR for the CA.
- Retrieves the CSR and enrolls it with a CA that supports issuance of subordinate CA certificates, such as Microsoft ADCS.
- Invokes the Vault API to install the signed CA certificate in the Vault PKI secrets engine. The certificate enables the CA to issue end-entity certificates via the HashiCorp role.
- Provisions the role information to the Vault. Only role information that was created by the Web SDK can provision to the vault.

To provision, you can use POST PKI/HashiCorp/CA in the *Developer's Guide*.

## Web SDK methods for the HashiCorp Vault PKI

Do not manually create the HashiCorp Vault PKI application objects. Instead, use the following Web SDK methods.

Some Web SDK methods require a {guid}. GUIDs uniquely identify a role, intermediate CA certificate, or HashiCorp application object. Be sure to specify use the correct context.

> **TIP** After an API call completes, you can view the result in the HashiCorp Vault and Policy Tree.

### Role methods

The Web SDK Role methods manage HashiCorp users that request certificates for end users.

- **Create policy for a HashiCorp Vault PKI role**: For more information, see POST PKI/HashiCorp/Role in the *Developer's Guide*.
- **Update policy for a HashiCorp Vault PKI role**: For more information, see PUT PKI/HashiCorp/Role/{guid} in the *Developer's Guide*.
- **View policy details of a HashiCorp Vault PKI role**: For more information, see GET PKI/HashiCorp/Role/{guid} in the *Developer's Guide*.
- **Delete policy for a HashiCorp Vault PKI role**: For more information, see DELETE PKI/HashiCorp/Role/{guid} in the *Developer's Guide*.

- **Enforce policy for roles in a HashiCorp Vault PKI secrets engine**: For more information, see POST PKI/HashiCorp/CA/{guid} in the *Developer's Guide*.

### CA methods

The Web SDK CA methods manage communication with the CA and Vault PKI secrets engine and role provisioning.

- **Create a HashiCorp Vault PKI application**: For more information, see POST PKI/HashiCorp/CA in the *Developer's Guide*.
- **Update a HashiCorp Vault PKI application**: For more information, see PUT PKI/HashiCorp/CA/{guid} in the *Developer's Guide*.
- **List HashiCorp Vault PKI applications**: For more information, see GET PKI/HashiCorp/CA in the *Developer's Guide*.
- **View details of a HashiCorp Vault PKI application**: For more information, see GET PKI/HashiCorp/CA/{guid} in the *Developer's Guide*.
- **Renew the CA certificate for a HashiCorp Vault PKI secrets engine**: For more information, see POST PKI/HashiCorp/CA/{guid}/Renew in the *Developer's Guide*.
- **Delete a HashiCorp Vault PKI application**: For more information, see DELETE PKI/HashiCorp/CA/{guid} in the *Developer's Guide*.

## Setting up for the HashiCorp Vault PKI application

Do not manually create the HashiCorp Vault PKI application objects. Start by setting up your HashiCorp Vault and completing the following high-level tasks using Certificate Manager - Self-Hosted web interface, or the Web SDK.

1. In HashiCorp, set up permissions. For more information, see "HashiCorp Vault permission requirements" on page 447.
2. In HashiCorp, enable the TLS option.
3. Create a Policy folder to hold the objects that will be created. Be sure to grant the API caller the required permissions. These permissions are required by the HashiCorp Vault PKI WebSDK methods.

4. Create a Password credential where the HashiCorp Vault token is the password. For help with creating credentials, see the *Administration Guide*.
5. Identify a CA template that is capable of issuing CA certificates. Example, the *SubCA* **Template** of a Microsoft CA.
6. Create credentials for the chosen CA template.
7. Create a CA template and test enrollment of a subordinate CA certificate.
8. (Optional) Create PKI roles by calling POST PKI/HashiCorp/Role. You can assign existing roles to HashiCorp Vault PKI applications at the time of creation or add them later.

## Creating and managing the HashiCorp PKI Vault

After you create a HashiCorp Vault PKI application and provision the role to HashiCorp, the Vault is ready. HashiCorp role users can begin requesting certificates via the Vault. The application manages communication with the CA.

### To create and manage HashiCorp PKI vault

1. Set up the HashiCorp Vault PKI. For more information, see https://www.vaultproject.io/intro/getting-started.
2. In HashiCorp, set up "HashiCorp Vault PKI roles" on page 447.
3. To create policy for a HashiCorp Vault PKI role, call POST PKI/HashiCorp/Role in the *Developer's Guide*.
4. To create a HashiCorp Vault PKI application and provision the role to HashiCorp, call POST PKI/HashiCorp/CA in the *Developer's Guide*.
5. (Optional) To manage roles or the application, use any of the Web SDK Methods for the HashiCorp Vault PKI in the *Developer's Guide*.

> **TIP** After an API call completes, you can view the result in the HashiCorp Vault and Policy Tree.

# IBM GSK configuration

The IBM Global Security Kit (GSK) is a library that can be integrated with server platforms such as IBM WebSphere MQ Server and Client, IBM HTTP Server (IHS), IBM WebSphere Application Server (WAS), and IBM Tivoli to enable SSL-protected communication.

GSK supports the CMS keystores.

GSK also provides a command line interface that can be used to generate, manage, and import certificates for supported keystores. In CyberArk Trust Protection Foundation™, the GSK application object provides the configuration parameters that Trust Protection Foundationrequires to manage certificates for supported keystores—either internally or remotely—using the GSK command-line interface.

Certificates and key pairs stored in these keystores can be consumed by a variety of GSK-integrated platforms offered by IBM.

The GSK application driver also supports provisioning certificates to IBM GSK's trust store.

> **NOTE** For information about the trust store, see About creating and configuring trust stores in the *CyberArk Trust Protection Foundation Certificate Management Guide*.

This section provides the information you need to manage certificates in GSK keystores.

This chapter contains the following topics:

## GSK certificate lifecycle

The IBM GSK driver generates and modifies keystores within Trust Protection Foundation when the following criteria are met:

- Keystores are in CMS format
- Certificates being provisioned aren't configured for remote key/CSR generation

**NOTE:** Remote generation with RSA 3072 and RSA 4096 bit keys is not supported.

The following table outlines the stages of the certificate lifecycle for certificates located in GSK keystores and the management level associated with each stage.

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| Stages 0-700 are performed by the GSK Application driver only if remote key generation is enabled. If the private key and CSR are locally generated on the Trust Protection Foundation server, stages 0-700 are performed by the X509Certificate Application driver.<br><br>The private key and CSR are remotely generated on the certificate's consumer application(s) if the **Generate Key/CSR on Application** option is enabled in the Certificate object. | | | | |
| 100 | CheckStore | No processing. | | |
| 200 | CreateConfigureStore | Only applies to remote generations.<br><br>Trust Protection Foundation creates a backup of the keystore if it exists. For more information, see "About keystore backups" on page 42. | | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| | | If the configured keystore does not exist and the **Create** option is selected on the GSK Application object, Trust Protection Foundation creates the certificate keystore. | | |
| 300 | CreateKey | No processing. | | |
| 400 | CreateCSR | Trust Protection Foundation creates the Certificate Signing Request (CSR). | x | x |
| 800 | CheckKeyStore | If the certificate and private key are generated on the Trust Protection Foundation server, this is the stage when Trust Protection Foundation determines if the configured keystore exists.<br><br>If the keystore does not exist and the **Create** option is selected on the GSK Application object, Trust Protection Foundation creates the certificate keystore.<br><br>If the keystore exists, Trust Protection Foundation creates a backup of the keystore on the same directory. For more information on the backup procedure, see "About keystore backups" on page 42. | | x |
| 801 | InstallCertificateChain | Trust Protection Foundation installs the root certificate chain. | | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| | | The certificate chain file is required to install the certificate.<br><br>Root certificates can be discovered and brought under management, or they can be manually imported. For more information, see Managing root certificates. | | |
| 802 | InstallCertificate | Trust Protection Foundation installs the certificate to the GSK keystore.<br><br>If the GSK driver cannot back up the keystore, it stops all certificate and private key operations in the keystore. | | x |
| 803 | ReplaceLabel | If the **Reuse Label** option is selected on the GSK Application object, Trust Protection Foundation re-installs the certificate with the correct label. | | x |

## Permission requirements

When Trust Protection Foundation provisions certificates or conducts an onboard Validation on a GSK keystore, the user account that Trust Protection Foundation uses to authenticate with the keystore must have the following permissions:

- Read and Write access to the keystore defined in the GSK Application object. For more information, see "Creating an IBM GSK application object" on page 457.

- Read and Write access to the Directory where the keystore is located. Trust Protection Foundation creates a backup of the keystore (*original_keystore_filename*- ved.bak) in the same Directory where the original keystore resides. For more information, see "About keystore backups" on page 42.

  > The GSK driver stops all certificate and private key operations in the keystore when it is unable to back up the keystore.

- Read and Write access to the Temp Directory defined in the Device object. For more information on the Device object configuration, see "Creating a device object in the Policy Tree" on page 239.

## GSK prerequisite configuration

To enable Trust Protection Foundation to provision certificates on Global Security Kit (GSK) keystores, you must perform the following configuration tasks:

1. Install the GSK software.

   Trust Protection Foundationsupports the following versions of the GSK utilities:

   - gsk7cmd
   - ikeycmd
   - gsk7capicmd
   - gsk8capicmd
   - GSK software is typically bundled with other IBM software.

2. Set up the keystore database.

   For more information, refer to your GSK utility documentation.

3. Grant Execute permissions to the java executable in the CLASS_PATH.

4. Grant the following permissions to the user account that Trust Protection Foundation uses to authenticate with the keystore:
   - Read and Write access to the Key Store Path defined in the GSK Application object. For more information, see "Creating an IBM GSK application object" on the facing page.
   - Read and Write access to the temp directory defined in the Device object. For more

information, see "Creating a device object in the Policy Tree" on page 239.

The user account must be an owner of the keystore.

5. Open the SSH port.

   Trust Protection Foundation uses the Secure Shell (SSH) protocol to manage certificates in GSK keystores; therefore, Trust Protection Foundation must have access to the server's SSH port. The default SSH port is port 22.

6. In the Policy Tree, create a Device object for the server where the keystore is installed. For more information, see "Creating a device object in the Policy Tree" on page 239.

7. In the Policy Tree, create and configure a GSK Application object.

   For more information on creating Application objects, see "Protecting server platforms and keystores" on page 223. For details on the object's settings, see "Creating an IBM GSK application object" below.

8. In the Policy Tree, associate the GSK Application object with the certificates stored in the keystore. For more information, see Associating a Certificate with an Application from the Certificate Object.

## About IBM GSK keystore backups

Whenever Trust Protection Foundation performs a certificate or private key operation, it creates a backup copy of the crypto file in the same directory where the original keystore resides.

If a backup operation fails, Trust Protection Foundation logs an event and halts processing. For example, if a backup operation fails for the JKS driver, the *JKS - Backup KeyStore Failed* event (ID 400E0020) is logged.

**CAUTION** If a CyberArk driver cannot back up a crypto file, it will not attempt any further certificate and private key operations.

## Creating an IBM GSK application object

To enable Trust Protection Foundation to manage supported keystore file formats using IBM Global Security Kit (GSK) standards, you must configure a GSK application object. This object provides the information Trust Protection Foundation needs to monitor, enroll, or provision GSK files.

**BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

**DID YOU KNOW?** When you add an installation to a certificate, you'll have the option of defining (and editing) this object during that process, which means that you don't have to log in to Policy Tree as the following procedure describes. And because the settings are the same, you can use this topic for information about each setting.

For more information, see Creating a certificate installation in Aperture in the *Certificate Management Guide*.

### To create and configure an IBM GSK application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **GSK**.
3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.
4. (Optional) In the **Device Certificate** box, click ... to select and associate a certificate with the new application.

   **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

   For detailed information on associating a certificate with an application object, see Associating a certificate with an application object.
5. Under **General**, do the following:
   a. In the **Application Name** field, type a name for the new application.
   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.

c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

Default system notifications are sent to the contact identities. The default contact is the master administrator.

> **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
>
> Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Under **Application Information**, do the following:

a. Click  next to **Application Credential** to browse for the credential object that you want to use to authenticate with the application.

> **DID YOU KNOW?** Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential might be a user name or private key credential; some drivers–such as F5, which is not SSH-based–can only use the user name credential for authentication.

> **NOTE** The user account you select must have *Read* and *Write* access to the Temporary, Private Key, and Certificate directories.
>
> If you need help with this step, see your system administrator.
>
> For more information, see Working with system credentials in the *Administration Guide*.

> **DID YOU KNOW?** The **Connection Method** is the protocol that Trust Protection Foundation uses to connect to the server and manage the certificates installed on that server. In an application object's settings, this field is typically read-only.

b. (Optional) In the **Port** field, type the port that Trust Protection Foundation should use to communicate with the server where the application is installed.

   Trust Protection Foundation uses the SSH protocol to communicate with the application server installed on Linux or Windows. The default SSH port assignment is port 22.

7. Complete the **Global Security Kit Settings** section by referring to the following table:

| Field | Policy | Description |
|---|---|---|
| **Global Security Kit Settings** | | |
| Version | | Version of the GSK you are using to manage GSK keystores.<br>Trust Protection Foundation supports the following versions:<br>◦ GSK 7.0<br>◦ GSK 8.0<br>Depending upon the version and key generation method selected, the GSK driver might require one of the following GSK utilties: gsk7cmd, ikeycmd, gsk7capicmd, and gsk8capicmd. |
| GSK Utility Path | | The file system path on the server where the GSK utilities are located. If this is left blank, the user's environment settings will be used. |
| Java Home Path | | The JAVA_HOME environment variable that is required to be set for the above utility path. If this is left blank, the user's environment settings will be used. |
| Key Store Path | | Full path to the keystore.<br>When you select a store type, Trust Protection Foundation automatically provides the keystore filename. However, you must define the full path. |
| Key Store Credential | | Password used to access the GSK keystore.<br>**IMPORTANT** The keystore password cannot contain the backslash character (\).<br>**To select a keystore credential**<br>a. Click ... to open the **Credential Selector** dialog.<br>b. Select the credential required to access the keystore, and then click **Select**. |

| Field | Policy | Description |
|---|---|---|
| | | If you select the **Create** option to create a new keystore, Trust Protection Foundation defines a new keystore password using the current credential.<br><br>Trust Protection Foundation does not include the keystore password on the command line when performing key management operations via the GSKCMD utility. Instead, it causes the utility to prompt for the password. |
| Create |  | Creates a new keystore file, if one does not already exist. |
| Replace Existing |  | Deletes the existing keystore and creates a new one.<br><br>This option is available only when the **Create** option is selected. Trust Protection Foundation automatically deselects this option after it replaces the keystore. |
| **Processing Options** | | The following are server-specific certificate settings. They are referenced only when you associate a certificate with the current GSK application object. |
| Certificate Label | | Label assigned to the key/certificate in the keystore so it can be used by the server or application.<br><br>This value must be unique within the keystore.<br><br>**NOTE** When certificates are discovered by the Server Agent, the agent captures and sends the certificate label to Trust Protection Foundation using the existing friendly name JSON object property. Trust Protection Foundation then sets the certificate label value when it creates the new application object.<br><br>**DID YOU KNOW?** Depending on the trust store, this setting goes by different names. For example, GSK refers to this setting as the *Certificate Label*, JKS refers to it as the *Certificate Alias*, and CAPI and PKCS#12 refer to it as the *Friendly Name*. |

| Field | Policy | Description |
|---|---|---|
| Reuse Label |  | Reuses the label that is assigned to the key/certificate in the keystore when the certificate is renewed. This option keeps the existing certificate available during the renewal process and simplifies management of the applications that use the key/certificate referenced by the label. |
| **CMS Options** | | The following options are available only for CMS keystores. |
| Use FIPS |  | Creates the keystore and manages certificates and private keys in accordance with the Federal Information Processing Standard (FIPS). Version 4 CMS keystores are considered FIPS compliant whereas Version 3 CMS keystores are not. If the keystore to which certificates are to be provisioned already exists, this setting must be consistent with the settings used when it was created. |
| Password Valid For |  | Number of days the password can be used to access the keystore. After this interval, the keystore is inaccessible.<br>The default value is 0 (password does not expire). |
| Stash Password |  | Generates an .sth file that stores the keystore password. Servers and applications reference the .sth file to obtain keystore passwords. |
| Default Certificate |  | Designates the current certificate as the default certificate in the keystore. |

8. (Optional) Under **File Ownership and Permissions**, select **Yes** on the **Set Owner and Permissions after Provisioning Files** drop-down—if you want to set specific permissions and ownership on files after they have been provisioned by Trust Protection Foundation—and then do the following:

   a. In the **Owner** field, type the user account name of the user who should have access to the provisioned files.

   > **BEST PRACTICE** Who you assign as owners and approvers of your certificates is an important part of your PKI strategy. This is especially true because employees continue to pose the greatest threat to securing trust. Typically, this is because many employees fail to follow security best practices.

b. From the **Owner Permissions** list, select the level of permissions you want to grant to the owner (*Read*, or *Read and Write*).

c. In the **Group** field, type the group name to which the owner belongs.

d. From the **Group Permissions** list, select the level of permissions you want to grant to that group (*None*, *Read*, or *Read and Write*).

9. When you are finished, click **Save**.

## What's next?

After you've created an application object, here are other things you can do to manage the new application:

- On the application's **Settings** sub-tab:
  - Click  to push a certificate to its associated application.

    For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.
  - Click  **Reset** to stop processing the application and reset the status and stage.
  - Click  to reattempt installation of the certificate to its associated application, .
  - Click  **Validate Now** to validate the applications associated certificate.

    Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

    For more information, see "About certificate and application validation" on page 270.
- On the application object's **Validation** tab, you can configure validation settings for the application object.
- On an object's **General** tab:

- Click the **Log** sub-tab to view any events that are triggered by the template object.
- Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

## IBM Sterling Connect:Directconfiguration

Many organizations use IBM Sterling Connect:Direct to securely transfer important files between their internal computer systems and external partners.

The Connect:Direct application driver installs certificates and private keys on devices hosting IBM Sterling Connect:Direct that have the optional Secure+ feature enabled. The driver also supports the provisioning of certificates to the Connect:Direct equivalent of a trust store.

> **NOTE** For information about the trust store, see About creating and configuring trust stores in the *CyberArk Trust Protection Foundation Certificate Management Guide*.

This section provides the information you need to correctly configure a Connect:Direct application object.

This chapter contains the following topics:

## Supported Versions of IBM Sterling Connect:Direct

CyberArk Trust Protection Foundation supports the following versions of IBM Sterling Connect:Direct:

- IBM Sterling Connect:Direct versions 6.0x, 6.1x, 6.2x, 6.3x, and 6.4x for Windows
- IBM Sterling Connect:Direct versions 6.0x, 6.1x, 6.2x, 6.3x, and 6.4x for UNIX

**IMPORTANT** In the newer versions of IBM Sterling Connect:Direct, certificates and keys are stored in a shared IBM CMS-formatted keystore. For older versions of IBM Sterling Connect:Direct, certificates and keys are stored in separate PEM files. This change in format resulted in some behavioral differences. While most changes are inconsequential, one key difference: because the CMS keystore format requires unique labels and a label can only be used once by each server that is running the newer versions of Connect:Direct.

## Connect:DirectPrerequisite Configuration

CyberArk Trust Protection Foundation™ communicates with Connect:Direct nodes using the Connect:Direct Application Interface for Java (CDAIJ). This requires the use of a dedicated Java client on the Trust Protection Foundation server. Because the Java client is the relying party securely connecting to the Connect:Direct node, you must verify that the client has the proper trust configuration to allow the connection.

**NOTE** This prerequisite configuration is for the driver and needs to be performed only once on each of the Trust Protection Foundation servers where you plan to do provisioning. This is different from prerequisite configuration required for the target device which has to be performed on every device to which Trust Protection Foundation provisions.

The trust store for the Java client is located in the *[path]*\Utilities\ConnectDirect directory. It is a standard JKS formatted keystore named *cacerts* with the standard password of *changeit*. To establish the proper trust relationship, use the Java keytool utility from a computer where the Java Runtime Engine (JRE) is installed to import root and intermediate certificates.

The default Java trust store is provided for example purposes under the name *cacerts.sample*. If all of the Connect:Direct nodes in your environment are using certificates issued by public certificate authorities, it might be sufficient to simply rename that file to *cacerts*.

The syntax for importing trust certificates using the keytool utility is similar to the following examples:

```
keytool -importcert -file root-ca.crt -alias root-ca -keystore cacerts -
storetype JKS -storepass changeit -trustcacerts
```

```
keytool -importcert -file issuer-ca.crt -alias issuer-ca -keystore cacerts -
storetype JKS -storepass changeit -trustcacerts
```

**IMPORTANT** After you complete the installation, you need to navigate to the ConnectDirect directory, and rename the `cacerts` file. By default this driver comes with a `cacerts.sample` file that need to be renamed to `cacerts`. This prevents your `cacerts` file from being overwritten during an upgrade (since a file named `cacerts` is not included with the driver). The file can be found in the following directory:

```
\Program Files\Venafi\Utilities\ConnectDirect\
```

## Connect:Direct Certificate Lifecycle

The following table outlines certificate lifecycle stages for certificates and private keys installed on an IBM Sterling Connect:Direct node and identifies the management level associated with each stage.

Lifecycle stages 800 through 1200 are handled by application drivers. However, not all stages are defined for every driver. For more information about lifecycle stages, see "About certificate lifecycle management" on page 42 in the *CyberArk Trust Protection Foundation Certificate Management Guide*.

For the Connect:Direct driver, stage 800 is defined in the following table.

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 800 | InstallCertificate | Installs the certificate, private key, and (optionally) the root and intermediate certificate chain. | |  |

## Connect:Direct Permission Requirements

The Username Credential used by the Trust Protection Foundation must be a Connect:Direct user authority which has been granted the user privilege necessary to configure Secure+.

For Windows systems, either the Secure+ parameter or the Admin parameter of the User Functional Authority must be set to *Yes* to enable remote execution of Secure+ commands.

**NOTE** Because members of the local Administrators group are granted the Admin user privilege implicitly, they do not need to be granted any privileges.

For Unix and Linux systems, you must set the cmd.s+conf parameter of the local user information record (in userfile.cfg) to a value of *y* to enable remote execution of Secure+ commands. Alternatively, if you do not set the cmd.s+conf value, then the Admin parameter must be set to *y*.

**IMPORTANT** The Connect:Direct driver has been designed to support connectivity over any cipher suite that is supported by Java 7 and that is allowed by the remote node. This includes several cipher-block chaining (CBC) cipher suites that have known vulnerabilities. For this reason, you should configure only non-CBC cipher suites on the remote node to which the Trust Protection Foundation connects. See http://www-01.ibm.com/support/docview.wss?uid=swg21598988 for more information about the vulnerability.

## Creating a Connect:Direct application object

To install an SSL certificate onto an IBM Sterling Connect:Direct node, you must apply the required configuration settings described in this topic.

**NOTE** Concurrency is not supported by the Connect:Direct driver; so the *Concurrent Connection Limit* setting of the device must be set to 1 to ensure reliable operation. Refer to your IBM Sterling Connect:Direct Application Interface for Java (CDAIJ) documentation.

**BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

### To create and configure a Connect:Direct application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **Connect:Direct**.
3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.

4. (Optional) In the **Device Certificate** box, click ... to select and associate a certificate with the new application.

   > **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

   For detailed information on associating a certificate with an application, see Associating a certificate with an application object.

5. Under **General**, do the following:

   a. In the **Application Name** field, type a name for the new application.

   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.

   c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

      Default system notifications are sent to the contact identities. The default contact is the master administrator.

      > **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
      >
      > Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

   d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

      The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Under **Application Information**, do the following:

a. Click  next to **Application Credential** to browse for the credential object that you want to use to authenticate with the application.

> **DID YOU KNOW?** Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential might be a user name or private key credential; some drivers–such as F5, which is not SSH-based–can only use the user name credential for authentication.

> **NOTE** The user account you select must have *Read* and *Write* access to the Temporary, Private Key, and Certificate directories.
>
> If you need help with this step, see your system administrator.
>
> For more information, see Working with system credentials in the *Administration Guide*.

b. Click the **Connection Method** list, click the protocol to use–HTTPS or SSH–and then in the **Port** field, specify the associated port number.

c. From the **API Protocol** drop-down, select either **SSL** or **TLS**.

This value must match the Secure+ Protocol configured under the Security Options of the Connect:Direct endpoint to which Trust Protection Foundation is provisioning.

d. (Optional) In the **Port** field, type the TCP port on which the Connect:Direct endpoint is listening for API connections.

The default API port assignment is 1363.

7. Complete the settings for the IBM Sterling Connect:Direct application object by referring to the following table:

| Field | Policy | Description |
|---|---|---|
| **Connect:Direct Settings** | | |
| Node Name | No | The name of the Connect:Direct Secure+ node to which Trust Protection Foundation provisions the certificate and private key. |
| Install Chain | Yes | This setting controls whether or not the root and intermediate certificate chain is installed with the certificate and private key when they are provisioned to the Connect:Direct Secure+ node. |
| Key Certificate Alias | No | An alias that represents the instance of the certificate and private key when they are assigned to the Connect:Direct Secure+ node. For current versions of Secure+ the alias is the name of the actual Key Certificate (PEM) file that contains the certificate and private key, though that behavior is ultimately dictated by the Connect:Direct API and the keystore format used by Secure+. |

8. When you are finished, click **Save**.

## What's next?

After you've created an application object, here are other things you can do to manage the new application:

- On the application's **Settings** sub-tab:
  - Click  to push a certificate to its associated application.

    For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.
  - Click  **Reset** to stop processing the application and reset the status and stage.
  - Click  to reattempt installation of the certificate to its associated application, .
  - Click  **Validate Now** to validate the applications associated certificate.

Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

For more information, see "About certificate and application validation" on page 270.

- On the application object's **Validation** tab, you can configure validation settings for the application object.
- On an object's **General** tab:
  - Click the **Log** sub-tab to view any events that are triggered by the template object.
  - Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

# IBM WebSphere DataPower

This section documents CyberArk Trust Protection Foundation's implementation of the IBM WebSphere DataPower application and provides the information you need to manage certificates on supported DataPower devices.

> **NOTE** Only Reverse (Server) Crypto Profile is supported for the DataPower Onboard Discovery and the provisioning driver.

The DataPower application driver also supports provisioning certificates to the IBM DataPower's trust store.

> **NOTE** For information about the trust store, see About creating and configuring trust stores in the *CyberArk Trust Protection Foundation Certificate Management Guide*.

This chapter contains the following topics:

## IBM WebSphere DataPower certificate lifecycle

The following table outlines the stages of the certificate lifecycle for certificates installed on supported DataPower devices and the management level associated with each stage.

**NOTE:** Remote generation with RSA 3072 bit keys is not supported.

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| Stages 0-700 are performed by the DataPower Application driver only if remote key generation is enabled. If the private key and CSR are locally generated on the Trust Protection Foundation server, stages 0-700 are performed by the X509Certificate Application driver.<br><br>The private key and CSR are remotely generated on the certificate's consumer application(s) if the **Generate Key/CSR on Application** option is enabled in the Certificate object. | | | | |
| 0 | StartProcessing | No processing. | | |
| 100 | CheckStore | No processing. | | |
| 200 | CreateConfigureStore | No processing. | | |
| 300 | CreatePrivateKeyandCSR | Trust Protection Foundation creates the private key and CSR. | x | x |
| 400 | RetrieveCSR | Trust Protection Foundation retrieves the certificate from the CA. | x | x |
| 800 | InstallPrivateKey | Trust Protection Foundation installs the private key if the private key was not created on the host. | | x |
| 801 | InstallCertificate | Trust Protection Foundation installs the certificate on the DataPower device. | | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 802 | Create Configured Objects and Associate with Certificate and Private Key | If the **Validation Credential** option is selected in the DataPower Application object configuration, Trust Protection Foundation creates a Validation Credential object using the currently installed certificate. For more information, see "Creating an IBM WebSphere DataPower application object" on page 479.<br><br>If the **Identification Credential** option is selected in the DataPower Application object configuration, Trust Protection Foundation creates an Identification Credential object using the currently installed certificate. For more information, see "Creating an IBM WebSphere DataPower application object" on page 479.<br><br>If the **Associate to Profile** option is selected in the DataPower Application object configuration, Trust Protection Foundation associates the certificate with an SSL Client Profile, SSL Server Profile and/or SSL Proxy Profile object as configured. For more information, see "Associate to Profile" on page 484. | | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| | | ■ If **Install Certificate Chain** is selected, Trust Protection Foundation installs the root and the intermediate certificates on the DataPower device. For more information, see "Install Certificate Chain" on page 486.<br><br>Root certificates can be discovered and brought under management, or they can be manually imported. For more information, see “Managing Root Certificates” on page 1. | | |
| 900 | CheckConfiguration | This stage is reserved for future use. | | |
| 1000 | ConfigureApplication | This stage is reserved for future use. | | |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 1100 | RestartApplication | This stage is reserved for future use. | | |
| 1200 | EndProcessing | Trust Protection Foundation completes the certificate processing and, if configured, runs a Validation check on the certificate and application. | | x |
| 1400 | Revocation | Trust Protection Foundation submits a revocation request to the CA.<br>Certificate revocation is a certificate operation; it does not involve the application driver. | x | x |

## IBM WebSphere DataPower prerequisite configuration

To enable Trust Protection Foundation to provision certificates on supported DataPower devices over an HTTPS connection, you must complete the following high-level tasks:

1. Enable the REST and XML Management interfaces on the DataPower Gateway (**Network** > **Management**).
2. Ensure that network connectivity exists between Trust Protection Foundation servers and the API interfaces of the DataPower Gateway (ports 5554 and 5550 by default). For example, you might need to open the firewall.
3. Ensure that the user account used by the DataPower driver to interact with the device has the least privilege access required for the provisioning mode you intend to use:

   For Basic provisioning mode:

   ```
   */*/config/rmi-view-details?Access=r+x
   ```

```
*/*/config/save-config?Access=x
*/*/crypto/cert?Access=r+w+a+d
*/*/crypto/crypto-export?Access=x
*/*/crypto/key?Access=r+w+a+d
*/*/file/cert?Access=r+w+a+d
*/*/file/sharedcert?Access=r+w+a+d
*/*/file/temporary?Access=r+d
```

For Advanced provisioning mode, which includes all of the Basic provisioning mode access, plus:

```
*/*/crypto/idcred?Access=r+w+a+d
*/*/crypto/profile?Access=r+w+a
*/*/crypto/ssl-client?Access=r+w+a
*/*/crypto/ssl-server?Access=r+w+a
*/*/crypto/sslproxy?Access=r+w+a
*/*/crypto/valcred?Access=r+w+a+d
```

Additionally, when using HTTPS connection mode:

```
*/*/login/rest-mgmt?Access=x
*/*/login/xml-mgmt?Access=x
```

Additionally, when using SSH connection mode:

```
*/*/login/ssh?Access=x
```

## About provisioning certificates using SSH

To enable Trust Protection Foundation to provision certificates on supported DataPower devices using SSH, you must complete the following high-level tasks:

1. Enable SSH access to the DataPower SSL module.

   To learn how to do this, visit:

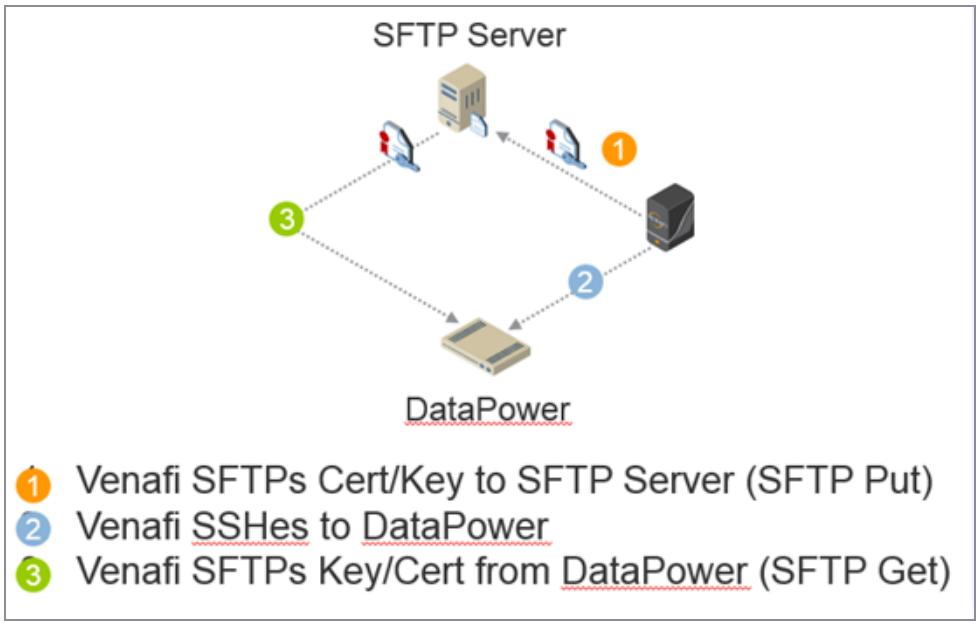
   http://www-01.ibm.com/support/docview.wss?uid=swg21469023.

2. Trust Protection Foundation uses the Secure Shell (SSH) protocol to manage certificates on supported DataPower devices.

3. Configure the SFTP staging server to receive connections from both Trust Protection Foundation and the DataPower device.

## Important information about the SFTP server

The DataPower appliance does not allow files to be uploaded directly to it. Files have to be downloaded by the device so once the needed files are staged on the SFTP server, Trust Protection Foundation logs in to the device and initiates the download. The SFTP server needs to always be *on* and available.




4. If necessary, open the firewall to allow SSH connections from Trust Protection Foundation to the DataPower device.

   The default SSH port is port 22.

5. In the Trust Protection FoundationPolicy Tree, create a Device object for the DataPower device.

   For more information, see "Managing device objects" on page 238.

6. In the Trust Protection FoundationPolicy Tree, create and configure an Application object for the DataPower device.

   For more information on creating application objects, see "Managing application objects" on page 259. For details on the object's settings, see "Creating an IBM WebSphere DataPower application object" on the facing page.

7. In the Trust Protection FoundationPolicy Tree, associate the DataPower Application object with the certificates installed on the DataPower device.

   For more information, see Associating a Certificate with an Application from the Certificate Object.

## Creating an IBM WebSphere DataPower application object

To enable Trust Protection Foundation to manage certificates and keys installed on DataPower Gateways, you must configure the DataPower application object. This object provides the information Trust Protection Foundation needs to monitor, enroll, or provision certificates and keys on its associated devices.

**BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

**DID YOU KNOW?** When you add an installation to a certificate, you'll have the option of defining (and editing) this object during that process, which means that you don't have to log in to Policy Tree as the following procedure describes. And because the settings are the same, you can use this topic for information about each setting.

For more information, see Creating a certificate installation in Aperture in the *Certificate Management Guide*.

### To create and configure an IBM DataPower application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **DataPower**.
3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.
4. (Optional) In the **Device Certificate** box, click  to select and associate a certificate with the new application.

   **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

For detailed information on associating a certificate with an application, see Associating a certificate with an application object.

5. Under **General**, do the following:

   a. In the **Application Name** field, type a name for the new application.

   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.

   c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

      Default system notifications are sent to the contact identities. The default contact is the master administrator.

      > **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
      >
      > Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

   d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

      The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

   e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

      You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

      For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Under **Application Information**, do the following:

a. Click  next to **Application Credential** to browse for the credential object that you want to use to authenticate with the application.

> **DID YOU KNOW?** Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential might be a user name or private key credential; some drivers–such as F5, which is not SSH-based–can only use the user name credential for authentication.

> **NOTE** The user account you select must have *Read* and *Write* access to the Temporary, Private Key, and Certificate directories.
>
> If you need help with this step, see your system administrator.
>
> For more information, see Working with system credentials in the *Administration Guide*.

b. (Optional) In the **Port** field, type the port that Trust Protection Foundation should use to communicate with the server where the application is installed.

   Trust Protection Foundation uses the SSH protocol to communicate with the application server installed on Linux or Windows. The default SSH port assignment is port 22.

c. (Optional) Type the **Port** used by the DataPower REST Management Interface on which the DataPower endpoint is listening for API connections.

7. Complete the settings for the application object by referring to the following table:

| Field | Policy | Description |
|---|---|---|
| **SFTP Configuration** | | Trust Protection Foundation imports certificates on DataPower devices via SFTP when Connection Method is set to SSH. |
| Hostname/Address |  | IP address or hostname of the SFTP server Trust Protection Foundation uses to install certificates on the DataPower device.<br>Trust Protection Foundation supports both IPv4 or IPv6 connections. |
| Credential |  | The user name Credential that Trust Protection Foundation uses to access the SFTP server.<br>**To select a user name credential**<br>a. Click  to open the **Credential Selector** dialog.<br>b. In the **Credential Selector** dialog box, select the **Username Credential** that Trust Protection Foundation must use to access the CA server, and then click **Select**.<br>For more information, see Managing System Credentials in the *CyberArk Trust Protection Foundation Administration Guide*. |
| Path |  | Directory on the SFTP host where Trust Protection Foundation can write (or read) certificate files for import (or export) on the DataPower device. |
| Port |  | Port Trust Protection Foundation uses to connect to the SFTP server.<br>The SFTP default port is 22. |
| **Configuration** | | The following settings are only relevant if the certificate associated with the current device is managed under the Provisioning level of certificate management. |

| Field | Policy | Description |
|---|---|---|
| Provisioning Mode: | | Choose either the *Advanced* (default) or *Basic* provisioning modes. |
| | *Advanced*: | This is the default setting and is designed to provision the certificate and then bind it to all of the configuration elements on the DataPower device.<br><br>**To configure Advanced provisioning mode**<br>a. From the **Provisioning Mode** list, select **Advanced**.<br>b. In the **Application Domain** field, specify the domain where the certificate and private key are installed on the DataPower device.<br>If nothing is specified, the default domain is used.<br>c. Complete the SSL Profile Configuration settings (described below).<br>See "SSL Profile Configuration" on the next page. |
| | *Basic*: | Installs the certificate and creates the crypto object that represents the certificate on the DataPower device. With this option, you'll need to perform all of the other DataPower configurations manually.<br><br>**To configure Basic provisioning mode**<br>a. From the **Provisioning Mode** list, select **Basic**.<br>b. In the **Crypto Certificate** field, type the name of your crypto certificate.<br>c. (Optional) In the **Crypto Key** field, type the name of your crypto key.<br>d. In the **Application Domain** field, specify the domain where the certificate and private key are installed on the device. |

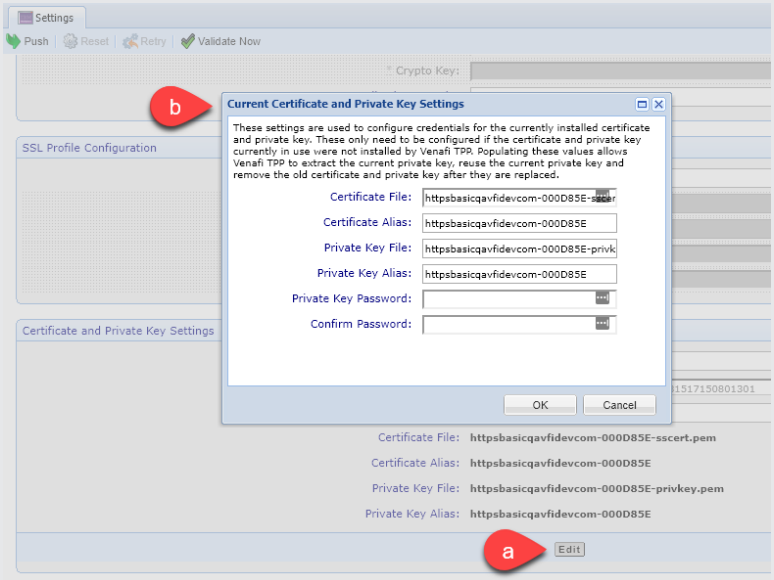
| Field | Policy | Description |
|---|---|---|
| | | If nothing is specified, the default domain is used. |
| **SSL Profile Configuration** | | To configure the SSL configuration settings via policy, go to the **Applications** > **DataPower** tab in the Policy object configuration. The DataPower settings defined in the Policy object may be inherited by all subordinate DataPower application objects. |
| Associate to Profile |  | Associates the specified profile(s) when Trust Protection Foundation provisions a certificate to the DataPower Gateway so that all SSL/TLS services associated with the DataPower Gateway can use the private key and certificate. |
| Credential Type |  | Select the type of credential to be created from the provisioned certificate:<br>◦ **Identification**: An identification credential consists of a certificate (which contains a public key) and the corresponding private key. The identification credential is used by the DataPower Gateway when it is an SSL proxy; that is, the SSL Proxy Service sends its certificate to an SSL peer when negotiating an SSL/TLS connection. The certificate may also be used as the identity when signing, encrypting, or decrypting documents.<br>◦ **Validation**: A validation credential is used to authenticate certificates that are received from SSL peers or to validate certificates that are used in digital signature and encryption operations. An SSL *client* requires a validation credential object only when it authenticates the certificate presented by the remote SSL server. An SSL *server* requires a validation credential object only when it authenticates remote SSL clients. The SSL standard does not require authentication of the server or client certificates. |

| Field | Policy | Description |
|---|---|---|
| Profile Type |  | Select the type of profile to which the certificate is bound:<br>◦ **SSL Proxy**: An SSL proxy profile is a legacy object that, in conjunction with a crypto profile, is used to secure inbound and/or outbound connections with the DataPower Gateway.<br>**TIP** SSL Proxy Profiles were deprecated in firmware version 7.2 but are still supported.<br>◦ **SSL Client**: An SSL client profile is used to secure outbound connections between the DataPower Gateway (acting as a client) and servers.<br>◦ **SSL Server**: An SSL server profile is used to secure inbound connections between clients and the DataPower Gateway (acting as a server).<br>For more information, visit https://www.ibm.com/support/knowledgecenter/en/SS9H2Y_7.5.0/com.ibm.dp.doc/sslconnections.html |
| Crypto Profile Name | | Specify the name of your Crypto Profile with which to associate the SSL Proxy Profile. |
| SSL Proxy Profile Name |  | Specify the name of the SSL Proxy Profile (deprecated) with which to bind the provisioned certificate. If the profile does not exist, it will be created. |
| SSL Client Profile Name |  | Specify the name of the SSL Client Profile with which to bind the provisioned certificate. If the profile does not exist, it will be created. |

| Field | Policy | Description |
|---|---|---|
| SSL Server Profile Name | | Specify the name of the SSL Server Profile with which to bind the provisioned certificate. If the profile does not exist, it will be created. |
| **Certificate and Private Key Settings** | | |
| Certificate Folder | | Specify the target folder for the certificate being provisioned.<br>The **cert:** option is the default certificate folder.<br>When you select the **sharedcert:** option, the certificate is provisioned into the default domain and the Application Domain setting applies to all other objects (Crypto Certificate, Crypto Key, Crypto Profile, etc.).<br>For more information about the DataPower requirements related to this setting, visit:<br>https://www.ibm.com/support/knowledgecenter/en/SS9H2Y_6.0.0/com.ibm.dp.xm.doc/filemanagement_directories.html |
| Install Certificate Chain | | Installs the root and intermediate root certificates required to support the signature chain with the current certificate. |
| Password Alias | | This IBM DataPower password map establishes a mapping between an alias and its corresponding plaintext password. The alias is a publicly known reference included in configuration files and reports. Leveraging this alias ensures the confidentiality of the actual password, as its plaintext form is securely stored in an encrypted file. Consequently, the password linked to the alias remains inaccessible and undisclosed to unauthorized individuals. |

| Field | Policy | Description |
|---|---|---|
| | | **NOTE** This feature will not keep in sync with passwords under Password Alias. Using a different "Private Key Password Credential" in Trust Protection Foundation doesn't update the existing Password Alias on DataPower.<br>This enhancement utilizes existing aliases in DataPower. A new Password Alias will be created if no alias exists and a "Private Key Password Credential" is provided in Trust Protection Foundation. |
| Private Key Password Credential |  | The credential that defines the password used to encrypt the private key before it is provisioned to the file on the DataPower Gateway.<br>**NOTE** This credential is not applicable when provisioning a validation credential in Advanced mode, or when provisioning only a crypto certificate in Basic mode.<br>**To select a private key password credential**<br>a. Click  to open the **Credential Selector** dialog.<br>b. Select the credential required to access the private key file for certificate renewal, and then click **Select**.<br>For more information, see Managing system credentials in the *CyberArk Trust Protection Foundation Administration Guide*. |
| Use FIPS |  | **NOTE** The Federal Information Processing Standard (FIPS) option for private key storage is not supported for DataPower in this release of Trust Protection Foundation. Please contact CyberArk Support with any questions or concerns. |
| Certificate File | | The filename Trust Protection Foundation generates automatically for the certificate when it installs the certificate.<br>This is not an editable field. |

| Field | Policy | Description |
|---|---|---|
| Certificate Alias | | Alias that is assigned to DataPower objects related to the certificate (i.e. Crypto Certificate, Identification Credential, and Validation Credential).<br><br>When using Advanced provisioning mode, this is not an editable field. |
| Private Key File | | The filename Trust Protection Foundation generates automatically for the private key file when it installs the private key.<br><br>This is not an editable field. |

| Field | Policy | Description |
|---|---|---|
| Private Key Alias | | Alias that is assigned to DataPower objects related to the private key (i.e. Crypto Key).<br><br>When using Advanced provisioning mode, this is not an editable field. |
| (Conditional) Private Key Password/Confirm | | The password required to access the private key.<br><br>This option is available only when you click **Edit** to enter the Certificate and Private Key settings for a certificate and private key not installed by Trust Protection Foundation.<br><br>  <br><br><br>a. Click **Edit** to open the **Current Certificate and Private Key Settings** box.<br><br>b. Type the settings for the currently installed certificate and private key, and then click **OK**.<br><br>These values need to be configured only if the current certificate and private key were not installed by Trust Protection Foundation. |

| Field | Policy | Description |
|---|---|---|
| | | Trust Protection Foundation uses the values to create Generational Credential objects for the currently installed certificate and private key. Populating these values allows Trust Protection Foundation to extract the current private key, reuse the current private key, and remove the old certificate and private key after they are replaced.<br><br>For more information on Generational credentials, see Managing System Credentials in the *CyberArk Trust Protection Foundation Administration Guide*. |

8. When you are finished, click **Save**.

## What's next?

After you've created an application object, here are other things you can do to manage the new application:

1. On the application's **Settings** sub-tab:
   - Click  to push a certificate to its associated application.

     For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.
   - Click  **Reset** to stop processing the application and reset the status and stage.
   - Click  to reattempt installation of the certificate to its associated application, .
   - Click  **Validate Now** to validate the applications associated certificate.

     Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

     For more information, see "About certificate and application validation" on page 270.

2. On the application object's **Validation** tab, you can configure validation settings for the application object.

3. On an object's **General** tab:

    - Click the **Log** sub-tab to view any events that are triggered by the template object.

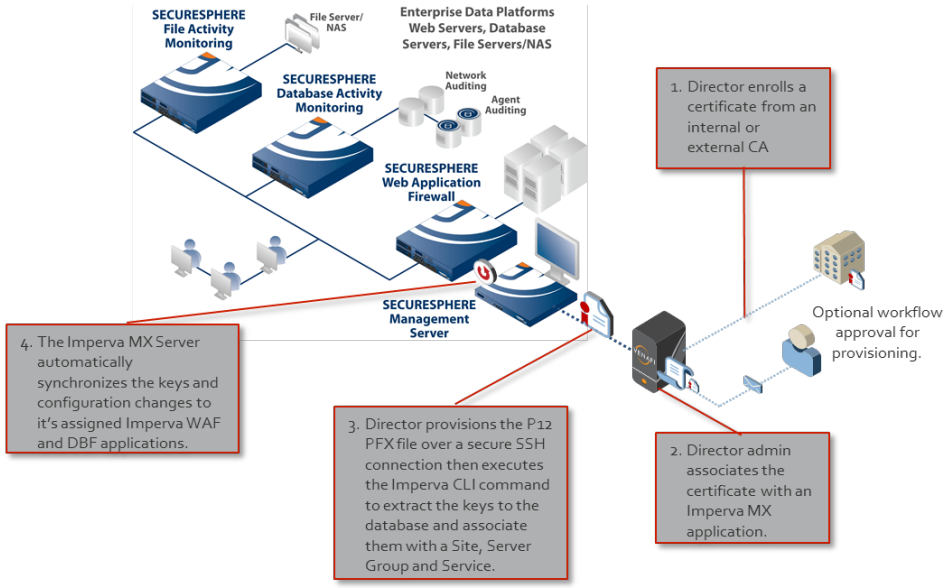
    - Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

# Imperva MX configuration

The CyberArk Trust Protection Foundation™ Imperva® MX driver supports provisioning of SSL certificate public and private key pairs to Imperva SecureSphere Management servers (MX) that synchronize to Imperva Web Application Firewalls (WAF) and Imperva Database Firewalls (DBF).

## Imperva MX Driver Overview

The Imperva MX Driver provisions a public and private SSL certificate key pair to an Imperva Management Server on SecureSphere, version 9.5.



The Imperva MX driver uses a P12 password protected PFX format and Imports the keys into the Imperva MX with the Imperva key tool using a temporary .csv file. The following attributes can also be configured in the driver and associated with the certificate key pair on the Imperva MX:

- Key Name
- Site
- Server Group
- Service
- HSM – True/False
- Temporary Random PFX filename
- Random Password for the PFX

Here is an example of the .csv content used to set the attributes for the key pair during provisioning:

```
www.qa.thing.net-446317BB00000012BF78-8BC8DB, qa.thing.net, tomcat, IIS, false, , ,
/tmp/randomfilename.pfx, randomPassword
```

These attributes coincide with the Imperva SecureSphere MX tree structure:

- **Site** (Folder folder: Datacenter or Location name)
- **Server Group** (One or more web server, referenced by IP, TCP port)
- **Service** (Web folders such as Apache, Tomcat, Websphere referenced by TCP port) / Application (Web app, referenced by URL or Hostname)

Here is another, more literal example:

- US-East
  - company.com
    - tomcat
    - public-site
    - partner-site

- webmail.company.com
  - IIS
  - OWA

- EMEA
  - company.de
    - tomcat
    - public-site

## File Naming

The Imperva MX driver manages the process of naming the keys as they are imported into the database. This is the best method for provisioning new and renewal certificate key pairs to the Imperva MX. This method is consistent with other Trust Protection Foundation provisioning drivers that control the file names. The public and private key pair is represented in the Imperva MX database with one key name. Automated key naming will be handled as follows:

Original common name:

**www.qa.somedepartment.myreallycoolradicalawesomedomainthing.net**

Trust Protection Foundation generated key name:

**www.qa.somedepartment.myreallycoolradicalawesomedomainthing.net-446317BB00000012BF78-8BC8DB**

Trust Protection Foundation also includes the certificate's serial number and a six digit hash of the serial number to ensure uniqueness.

The Imperva MX has a 200 character file name limitation so it is doubtful any truncation will ever be needed for the common name. In the event it is needed the common name would be truncated from the permission most characters.

## Known Limitations - No Post Provisioning Validation

Currently there is no Imperva CLI command to query the database that can be used in Trust Protection Foundation's post provisioning process for onboard or network validation (compares what is on the Imperva database with the certificate in the Trust Protection Foundation's secret store.) The workaround for now is

to turn off network and onboard validation in the Imperva MX driver settings.Venafi has requested from Imperva a CLI or API command to allow for validation; there is currently no availability date or commitment at this time.

This chapter contains the following topics:

## Imperva Certificate Key Pair Provisioning

As shown in the table below, there is only one stage for certificate key pair provisioning for Imperva SecureSphere MX applications.

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| Stages 0-700 are performed by drivers only if remote key generation is enabled and available on the application; the Imperva SecureSphere Management Server does not have this feature available. The private key and CSR are locally generated on the Trust Protection Foundation server, so stages 0-700 are performed by the X509Certificate Application driver. | | | | |
| 0 | StartProcessing | No processing. | | |
| 800 | InstallKey | Trust Protection Foundation uses the key tool to import the associated SSL certificate's public and private key into the MX database with attributes to set the:<br>■ Key Name<br>■ Site<br>■ Server Group<br>■ Service<br>■ HSM - True/False<br>■ Temporary PFX filename<br>■ Random Password for the PFX | | x |

## Imperva Permission Requirements

When Trust Protection Foundation provisions a certificate key pair to an Imperva SecureSphere Management Server, the user account that Trust Protection Foundation uses to authenticate with the appliance must provide credentials for the SSH connection and for the application user permissions to execute command line options.

## Imperva prerequisite configuration

This driver was tested with Imperva SecureSphere, version 12.5, using the ExternalAPIClient.jar key tool.

**IMPORTANT** Before implementing CyberArk's Imperva driver, carefully review and complete the following prerequisite tasks.

### To enable Trust Protection Foundation to provision certificate keys to an Imperva SecureSphere Management Server

1. Using the Imperva SecureSphere Management Server console, set up a CyberArk or Trust Protection Foundation user with permissions to import keys into the *Site*, *Group* and *Service* objects.
2. Set up a CLI user with permissions to execute the ExternalAPIClient.jar command line key tool from an SSH session.
3. In the SecureSphere console, verify the Site, Group and Service objects are defined that will be named in the Trust Protection Foundation Imperva MX driver configuration.

   For more information, refer to your Imperva documentation.

   Imperva devices have the option to use a FIPS-140 integrated HSM (hardware security module). However, this Trust Protection Foundation Imperva MX driver does not have FIPS options at this time. If desire to store private keys on your Imperva in an HSM, please contact CyberArk customer support so that they can log an enhancement request for this feature.
4. Before completing the final steps below in Trust Protection Foundation, perform the following steps to disable Imperva's CSRF:
   a. From the MX CLI, navigate to /opt/SecureSphere/server/SecureSphere/jakarta-tomcat-secsph/webapps/SecureSphere/WEB-INF/.
   b. Open the bootstrap.properties file and add the following string:

```
client.include.test.cpt=false
```

c. Save your changes, and then reboot the MX.

> **DID YOU KNOW?** Why do I need to temporarily disable CSRF?
>
> Imperva has implemented a cross-site request forgery (CSRF) protection mechanism that is incompatible with the ExternalAPIClient.jar that the CyberArk Imperva driver relies upon to provision certificates to devices. Therefore, to enable our integration, the CSRF protection must be disabled.
>
> If you want to request that Imperva update this functionality so that it does not interfere with certificate lifecycle automation, you can submit a request to them directly by visiting https://www.imperva.com/sign_in.asp?retURL=/articles/Procedure/How-to-create-Feature-request.

5. From the Certificate Manager - Self-Hosted menu bar, click From the Platform menu bar, click Policy Tree.

6. From the **Credentials** tree, do the following:
   a. Configure the CLI user credentials.

   b. Configure the SecureSphere application user credentials.

## Creating an Imperva MX application object

This section describes the configuration settings needed to enable Trust Protection Foundation to install an SSL certificate key pair on an Imperva SecureSphere Management Server.

You can also provision certificates in bulk to Imperva using the sample Adaptable Bulk Provisioning PowerShell script. See Using sample Adaptable Bulk Provisioning PowerShell scripts in the Certificate Management Guide.

> **BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

### To create and configure an Imperva MX application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **Imperva**.
3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.
4. (Optional) In the **Device Certificate** box, click  to select and associate a certificate with the new application.

   > **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

   For detailed information on associating a certificate with an application, see Associating a certificate with an application object.
5. Under **General**, do the following:
   a. In the **Application Name** field, type a name for the new application.
   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.
   c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

      Default system notifications are sent to the contact identities. The default contact is the master administrator.

> **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
>
> Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

   The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

   You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

   For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Under **Application Information**, do the following:

   a. Click  next to **Application Credential** to browse for the credential object that you want to use to authenticate with the application.

   > **DID YOU KNOW?** Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential might be a user name or private key credential; some drivers–such as F5, which is not SSH-based–can only use the user name credential for authentication.

   > **NOTE** The user account you select must have *Read* and *Write* access to the Temporary, Private Key, and Certificate directories.
   >
   > If you need help with this step, see your system administrator.
   >
   > For more information, see Working with system credentials in the *Administration Guide*.

> **DID YOU KNOW?** The **Connection Method** is the protocol that Trust Protection Foundation uses to connect to the server and manage the certificates installed on that server. In an application object's settings, this field is typically read-only.

b. Click the **Connection Method** list, click the protocol to use–HTTPS or SSH–and then in the **Port** field, specify the associated port number.

c. (Conditional) In the **SSH Port** field, specify the port number that Trust Protection Foundation should use to communicate with the appliance via an SSH connection.

   The default SSH port assignment is 22.

d. (Optional) In the **Port** field, type the port that Trust Protection Foundation should use to communicate with the server where the application is installed.

   Trust Protection Foundation uses the SSH protocol to communicate with the application server installed on Linux or Windows. The default SSH port assignment is port 22.

7. Complete the settings for the application object by referring to the following table:

| Field | Policy | Description |
|---|---|---|

8. When you are finished, click **Save**.

## What's next?

After you've created an application object, here are other things you can do to manage the new application:

- On the application's **Settings** sub-tab:
  - Click  to push a certificate to its associated application.

    For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.
  - Click  **Reset** to stop processing the application and reset the status and stage.
  - Click  to reattempt installation of the certificate to its associated application, .
  - Click  **Validate Now** to validate the applications associated certificate.

Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

For more information, see "About certificate and application validation" on page 270.

- On the application object's **Validation** tab, you can configure validation settings for the application object.
- On an object's **General** tab:
  - Click the **Log** sub-tab to view any events that are triggered by the template object.
  - Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

## JKS Configuration

The Java Keystore (JKS) is a library that can be integrated with the JKS-integrated server platforms to enable SSL-protected communication. JKS also provides a Keytool utility that can be used to generate, manage, and import certificates for JCEKS and JKS keystores.

In CyberArk Trust Protection Foundation™, the JKS Application object provides the configuration parameters that Trust Protection Foundation requires to manage certificates for JCEKS and JKS keystores using the Java Keytool. Certificates and key pairs stored in these keystores can be consumed by JKS-integrated server platforms.

Some of the most common Java application servers include:

- GlassFish
- JBoss
- Jetty
- Tomcat
- WebLogic

**NOTE** For information about the trust store, see About creating and configuring trust stores in the *CyberArk Trust Protection Foundation Certificate Management Guide*.

This section provides the information you need to manage certificates in Java keystores.

This chapter contains the following topics:

## JKS certificate lifecycle

The following table outlines the stages of the certificate lifecycle for certificates located in Java keystores and the management level associated with each stage.

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| Stages 0-700 are performed by the JKS application driver only if remote key generation is enabled. If the private key and CSR are locally generated on the Trust Protection Foundation server, stages 0-700 are performed by the X509Certificate application driver.<br><br>**NOTE** The private key and CSR are remotely generated on the certificate's consumer application (s) if the **Generate Key/CSR on Application** option is enabled in the certificate object. | | | | |
| 0 | StartProcessing | No processing. | | |
| 100 | CheckStore | No processing. | | |
| 200 | CreateConfigureStore | Only applies to remote generation.<br><br>Trust Protection Foundation creates a backup of the keystore, if it exists.<br><br>If the configured keystore does not exist and the **Create** option is selected on the JKS application object, Trust Protection Foundation creates the certificate keystore.<br><br>For more information, see the "Create" on page 517 setting in the "Java Keystore Settings" on page 515 table. | |  |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 300 | CreateKey | No processing.<br><br>**DID YOU KNOW?** Stage 300 is used for key generation only when the API of a target device separates keypair and CSR generation. When they're combined, both key and CSR generation are always done at stage 400. | | |
| 400 | CreateCSR | Trust Protection Foundation creates the Certificate Signing Request (CSR).<br><br>If remote key generation is enabled, the Java keytool utility is used to generate the CSR. Otherwise, Trust Protection Foundation generates the CSR. | ✓ | ✓ |
| 500 | PostCSR | Trust Protection Foundation submits the CSR to the certificate authority (CA).<br><br>**NOTE** If you post a manual CSR, this is the first stage of the certificate lifecycle. | ✓ | ✓ |
| 600 | ApproveRequest | Trust Protection Foundation approves the certificate renewal at the CA. | ✓ | ✓ |
| 700 | RetrieveCertificate | Trust Protection Foundation retrieves the certificate from the CA. | ✓ | ✓ |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 800 | CheckKeyStore | If the certificate and private key are generated on the Trust Protection Foundation server, this is the stage when Trust Protection Foundation determines if the configured keystore exists.<br><br>If the keystore does not exist and the **Create** option is selected on the JKS application object, Trust Protection Foundation creates the certificate keystore.<br><br>If the keystore exists, Trust Protection Foundation creates a backup of the keystore on the same Directory.<br><br>For more information on the backup procedure, see "About keystore backups" on page 42. | |  |
| 801 | InstallCertificateChain | Trust Protection Foundation installs the root certificate chain.<br><br>**NOTE** The Java Keytool will not allow a certificate to be installed if the appropriate root and intermediate root certificates are not already installed in the keystore. | |  |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| | | **NOTE** Root certificates can be discovered and brought under management, or they can be manually imported. For more information, see Managing root certificates. | | |
| 802 | InstallCertificate | If remote key generation is enabled, Trust Protection Foundation uses the Java keytool utility to import the certificate reply obtained at stage 700.<br><br>Otherwise, Trust Protection Foundation imports both the private key (stage 300) and the certificate reply (stage 700) into the keystore.<br><br>For additional information, see "About the benefits of central key generation" on page 509. | |  |
| 803 | ReplaceLabel | If the **Reuse Label** option is selected on the JKS application object, Trust Protection Foundation re-installs the certificate with the correct label. | |  |
| 900 | CheckConfiguration | Reserved for future use. | | |
| 1000 | ConfigureApplication | Reserved for future use. | | |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 1100 | RestartApplication | Reserved for future use. | | |
| 1200 | EndProcessing | Trust Protection Foundation completes the certificate processing and, if configured, runs a Validation check on the certificate and application. | |  |
| 1400 | Revocation | Trust Protection Foundation submits a revocation request to the CA.<br><br>**NOTE** Certificate revocation is a certificate operation; it does not involve the application driver. |  |  |

## JKS Permission requirements

When Trust Protection Foundation provisions certificates or conducts an onboard Validation on a Java keystore, the user account that Trust Protection Foundation uses to authenticate with the keystore must have the following permissions:

- Read and write access to the keystore defined in the JKS Application object. For more information, see "Java Keystore Settings" on page 515.
- Read and write access to the Directory where the keystore is located. Trust Protection Foundation creates a backup of the keystore (*original_keystore_filename*-ved.bak) in the same Directory where the original keystore resides.
- Read and write access to the Temp Directory defined in the Device object. For more information on the Device object configuration, see "Creating a device object in the Policy Tree" on page 239.

## JKS prerequisite configuration

To enable Trust Protection Foundation to provision certificates on Java keystores, you must complete some or all of the following high-level tasks. However, it depends on whether you're using central or remote generation.

For additional information, see "About the benefits of central key generation" on the facing page.

**IMPORTANT** Items that begin with "(Conditional)" are only required when doing *remote* key generation.

1. (Conditional) If you're doing remote key generation, install Java Keytool utility.

   Trust Protection Foundation supports the following versions of the Java Keytool utility:

   - Java 1.6
   - Java 1.7
   - Java 1.8

2. (Optional) If you don't want the CyberArk driver to set up the keystore for you automatically, then set up the keystore database manually.

   If you choose to set up the database manually, refer to your Java Keytool utility documentation for additional information.

3. (Conditional) If you're doing remote key generation, grant *Execute* permissions to the java.bin executable in the CLASS_PATH.

   This is a Java Keytool dependency.

4. Grant the following permissions to the user account that Trust Protection Foundation uses to authenticate with the keystore:

   - Read and Write access to the Key Store Path defined in the JKS Application object. For more information, see "Java Keystore Settings" on page 515.
   - Read and Write access to the temp Directory defined in the Device object. For more information, see "Creating a device object in the Policy Tree" on page 239.

     The user account must be an owner of the keystore.

5. Open the SSH port.

Trust Protection Foundation uses the Secure Shell (SSH) protocol to manage certificates in Java keystores; therefore, Trust Protection Foundation must have access to the server's SSH port. The default SSH port is port 22.

6. In the Policy Tree, create a Device object for the server where the Java keystore is installed.

   For more information, see "Creating a device object in the Policy Tree" on page 239.

7. In the Policy Tree, create and configure a JKS application object.

   For more information on creating application objects, see "Creating an application" on page 265. For details on the object's settings, see "Creating a JKS application object" on the next page.

8. In the Policy Tree, associate the JKS application object with the certificates stored in the keystore.

   For more information, see Associating a certificate with an application from the certificate object.

9. If you want to configure remote key generation using an HSM, you must activate Advanced Key Protect, an optional add-on feature to CyberArk Trust Protection Foundation. For more information, see Advanced Key Protect in the *Trust Protection Foundation Administration Guide*.

## About the benefits of central key generation

A primary advantage of central key generation is that it eliminates the dependency on device-side utilities, such as Java Keytool. Typically, central generation is also faster because command line utilities are rarely optimized for performance.

In addition, central generation keeps a copy of the private key in Trust Protection Foundation to allow for key recovery, for provisioning the same certificate to multiple devices, and for security-related cases where you want to enable TLS traffic inspections.

When performing central generation, CyberArk's JKS driver places a temporary copy of the keystore from the managed system into memory so that Trust Protection Foundation can perform the key operations.

Whenever the JKS driver modifies the keystore—this happens at the installation step when the **Generate Key/CSR on Application** option is disabled (set to No)—it verifies that the keystore on the managed system has not changed. If the local keystore has not changed, the JKS driver overwrites the local copy with the updated copy of the keystore.

If the JKS driver detects that the local keystore on the managed system has been altered while it was modifying its copy of the keystore—for example, if someone manually added a key or certificate—then it returns an error stating that the "Keystore is in use by another process" and it halts processing on the

application without uploading or replacing the keystore, while not overwriting the local copy. The JKS driver also logs the “Certificate Chain Install Failed” (event ID 400E000F) and “Certificate Install Failed” (event ID 400E000B) events to the Trust Protection Foundation log store.

## About JKS keystore backups

Whenever Trust Protection Foundation performs a certificate or private key operation, it creates a backup copy of the crypto file in the same directory where the original keystore resides.

If a backup operation fails, Trust Protection Foundation logs an event and halts processing. For example, if a backup operation fails for the JKS driver, the *JKS - Backup KeyStore Failed* event (ID 400E0020) is logged.

**CAUTION** If a CyberArk driver cannot back up a crypto file, it will not attempt any further certificate and private key operations.

## Creating a JKS application object

To enable Trust Protection Foundation to manage certificates for JCEKS and JKS keystores, you must configure a JKS application object. This object provides the information Trust Protection Foundation needs to monitor, enroll, or provision certificates for JCEKS and JKS keystores.

**BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

**DID YOU KNOW?** When you add an installation to a certificate, you'll have the option of defining (and editing) this object during that process, which means that you don't have to log in to Policy Tree as the following procedure describes. And because the settings are the same, you can use this topic for information about each setting.

For more information, see Creating a certificate installation in Aperture in the *Certificate Management Guide*.

### To create and configure a JKS application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **JKS**.
3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.
4. (Optional) In the **Device Certificate** box, click ... to select and associate a certificate with the new application.

   > **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

   For detailed information on associating a certificate with an application, see Associating a certificate with an application object.
5. Under **General**, do the following:
   a. In the **Application Name** field, type a name for the new application.
   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.
   c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

      Default system notifications are sent to the contact identities. The default contact is the master administrator.

> **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
>
> Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

   The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

   You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

   For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Under **Application Information**, do the following:

   a. Click  next to **Application Credential** to browse for the credential object that you want to use to authenticate with the application.

   > **DID YOU KNOW?** Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential might be a user name or private key credential; some drivers–such as F5, which is not SSH-based–can only use the user name credential for authentication.

   > **NOTE** The user account you select must have *Read* and *Write* access to the Temporary, Private Key, and Certificate directories.
   >
   > If you need help with this step, see your system administrator.
   >
   > For more information, see Working with system credentials in the *Administration Guide*.

> **DID YOU KNOW?** The **Connection Method** is the protocol that Trust Protection Foundation uses to connect to the server and manage the certificates installed on that server. In an application object's settings, this field is typically read-only.

b. (Conditional) In the **SSH Port** field, specify the port number that Trust Protection Foundation should use to communicate with the appliance via an SSH connection.

   The default SSH port assignment is 22.

c. (Optional) In the **Port** field, type the port that Trust Protection Foundation should use to communicate with the server where the application is installed.

   Trust Protection Foundation uses the SSH protocol to communicate with the application server installed on Linux or Windows. The default SSH port assignment is port 22.

7. Under **Remote Generation Settings**, click the **Private Key Location** list and select where you want remotely generated key pairs to be created:

> **TIP** The Thales SafeNet Luna SA HSM and Entrust nShield HSM settings are not visible if you have not activated Advanced Key Protect. For more information, see Enabling Advanced Key Protect.

   - **Device**: Key pairs are generated on the device. This is the default setting.
   - **Thales SafeNet Luna SA HSM**: Key pairs are generated on a Thales SafeNet Luna SA HSM. When you select this option, do the following:
     - In the **Slot Number** field, type the HSM slot number used by your JKS.

       The slot is actually the only contents of the JKS file. If the JKS file doesn't exist, the Trust Protection Foundation JKS driver creates it using the specified slot number. If the JKS file already exists, the driver simply verifies that the slot specified in the file matches the Trust Protection Foundation configuration.
   - **Entrust nShield HSM**: Key pairs are generated on an Entrust nShield HSM. When you select this option, do the following:
     - In the **Java Vendor** field, select the vendor of the JDK you have installed on the device (either Oracle or IBM).
     - In the **Protection Type** field, select the appropriate level of protection:
       - **Module**: This is the baseline protection level. It requires that your device has been properly configured to use the HSM for key generation.

If you select this protection type, the Key Store Credential and Private Key Credential values are required.

- **Softcard**: This is a higher level of protection and is a kind of password that is stored on your HSM.

  If you selected Softcard as the Protection Type, then in the **Softcard Identifier** field, enter your softcard's 40-character hash.

  Selecting this option requires that the device has been properly configured to use the HSM for key generation and that a softcard has been previously generated using the HSM and that the requester knows the passphrase for that softcard.

8. Refer to the following table to complete the remaining settings:

| Field | Policy | Description |
|---|---|---|
| **Java Keystore Settings** | | |
| Keytool Path |  | Full path name of the Java keytool utility.<br>A full path includes the path *and* the filename. For example: `/bin/keytool`.<br>Beginning with Trust Protection Foundation 15.4, the Keytool Path only applies when **Generate Key/CSR on Application** is enabled on the certificate object. |
| Version |  | Version of the Java Keytool utility you are using to manage keystores.<br>Trust Protection Foundation supports the following versions of the Java Keytool utility:<br>◦ Java 1.6<br>◦ Java 1.7<br>◦ Java 1.8 |
| Store Type |  | Type of store managed via the current JKS Application object.<br>The store type determines the key file format. You must select the keystore type supported by the platforms and applications that consume the keystore's certificates.<br>Trust Protection Foundation supports the following keystore types:<br>◦ JCEKS<br>◦ JKS |

| Field | Policy | Description |
|---|---|---|
| Pkcs#12 Encryption Algorithm |  | You can choose the encryption algorithm used when exporting PKCS#12 files. Stronger algorithms are usually less compatible, especially with older software. |
| **Keystore** | | To configure these settings via policy, go to the **Applications** > **JKS** tab in the policy object's configuration.<br>The JKS keystore settings defined in the Policy object may be inherited by all subordinate JKS application objects. |
| Keystore Path |  | Full path to the keystore.<br>A full path includes the path *and* the filename. For example: `/opt/pki/keystore.jks` |
| Keystore Credential |  | Password used to access the keystore.<br>a. Click the **Browse** button to open the Credential Selector dialog.<br>b. Select a Password Credential object, and then click **Select**.<br>If you select the **Create** option to create a new keystore, Trust Protection Foundation defines a new keystore password using the current credential.<br>Trust Protection Foundation does not include the key store password on the command line when performing key management operations via the Java Keytool utility. Instead, it causes the utility to prompt for the password. |
| Private Key Credential |  | The credential required to access the private key file for certificate renewal.<br>**To select a private key password credential** |

| Field | Policy | Description |
|---|---|---|
| | | a. Click ... to open the **Credential Selector** dialog.<br>b. Select the credential required to access the private key file for certificate renewal, and then click **Select**.<br>For more information, see Managing system credentials in the *CyberArk Trust Protection Foundation Administration Guide*. |
| Create | | Creates a new keystore file, if one does not already exist. |
| Replace Existing | | Deletes the existing keystore and creates a new one.<br>This option is available only when the **Create** option is selected. Trust Protection Foundation automatically deselects this option after it replaces the keystore. |
| **Processing Options** | | The following are server-specific certificate settings. They are referenced only when you associate a certificate with a current JKS application object. |

| Field | Policy | Description |
|---|---|---|
| Certificate Alias | | A label that is assigned to the key/certificate in the keystore so it can be used by servers and applications accessing the Java Keystore.<br><br>This value must be unique within the keystore.<br><br>**NOTE** When certificates are discovered by the Server Agent, the agent captures and sends the certificate label to Trust Protection Foundation using the existing friendly name JSON object property. Trust Protection Foundation then sets the certificate label value when it creates the new application object.<br><br>**DID YOU KNOW?** Depending on the trust store, this setting goes by different names. For example, GSK refers to this setting as the *Certificate Label*, JKS refers to it as the *Certificate Alias*, and CAPI and PKCS#12 refer to it as the *Friendly Name*. |
| Reuse Alias |  | Reuses the alias that is assigned to the key/certificate in the keystore when the certificate is renewed. This option keeps the existing certificate available during the renewal process and simplifies management of the applications that use the key/certificate referenced by the alias. |
| Key Algorithm |  | Algorithm used to generate the key for the current certificate.<br><br>Trust Protection Foundation supports RSA encryption with key sizes of 512, 1024, 2048, and 4096. |

9. (Optional) Under **File Ownership and Permissions**, select **Yes** on the **Set Owner and Permissions after Provisioning Files** drop-down–if you want to set specific permissions and ownership on files after they have been provisioned by Trust Protection Foundation–and then do the following:

    a. In the **Owner** field, type the user account name of the user who should have access to the provisioned files.

> **BEST PRACTICE** Who you assign as owners and approvers of your certificates is an important part of your PKI strategy. This is especially true because employees continue to pose the greatest threat to securing trust. Typically, this is because many employees fail to follow security best practices.

b. From the **Owner Permissions** list, select the level of permissions you want to grant to the owner (*Read*, or *Read and Write*).

c. In the **Group** field, type the group name to which the owner belongs.

d. From the **Group Permissions** list, select the level of permissions you want to grant to that group (*None*, *Read*, or *Read and Write*).

10. When you are finished, click **Save**.

For additional HSM configuration information, see "Managing applications using HSM-protected keys and Advanced Key Protect" on page 271 and "Configuring HSM-based remote key generation" on page 273.

## What's next?

After you've created an application object, here are other things you can do to manage the new application:

- On the application's **Settings** sub-tab:
  - Click  to push a certificate to its associated application.

    For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.
  - Click  **Reset** to stop processing the application and reset the status and stage.
  - Click  to reattempt installation of the certificate to its associated application, .
  - Click  **Validate Now** to validate the applications associated certificate.

    Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

    For more information, see "About certificate and application validation" on page 270.

- On the application object's **Validation** tab, you can configure validation settings for the application object.
- On an object's **General** tab:
  - Click the **Log** sub-tab to view any events that are triggered by the template object.
  - Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

# iPlanet configuration

The CyberArk Trust Protection Foundation™ iPlanet driver supports iPlanet web server versions 6.1 and 7.0. This section provides the information you need to manage certificates on supported iPlanet servers.

This chapter contains the following topics:

## Certificate lifecycle

The following table outlines the stages of the certificate lifecycle for certificates installed on iPlanet servers and the management level associated with each stage.

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| Stages 0-700 are performed by the iPlanet application driver only if remote key generation is enabled. If the private key and CSR are locally generated on the Trust Protection Foundation server, stages 0-700 are performed by the X509Certificate application driver.<br><br>The private key and CSR are remotely generated on the certificate's consumer application(s) if the **Generate Key/CSR on Application** option is enabled in the Certificate object. | | | | |
| 0 | StartProcessing | No processing. | | |
| 100 | CheckStore | No processing. | | |
| 200 | CreateConfigureStore | No processing. | | |
| 300 | CreateKey | Trust Protection Foundation creates the private key. | x | x |
| 400 | CreateCSR | Trust Protection Foundation creates the Certificate Signing Request (CSR). | x | x |
| 500 | PostCSR | Trust Protection Foundation submits the CSR to the Certificate Authority (CA).<br><br>If you post a manual CSR, this is the first stage of the certificate lifecycle. | x | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 600 | ApproveRequest | Trust Protection Foundation approves the certificate renewal at the CA. | x | x |
| 700 | RetrieveCertificate | Trust Protection Foundation retrieves the certificate from the CA. | x | x |
| 800 | InstallCertificateChain | Trust Protection Foundation installs the root and intermediate certificates on the iPlanet server. | | x |
| 801 | InstallCertificate | Trust Protection Foundation installs the certificate on the iPlanet server. | | x |
| 900 | CheckConfiguration | Reserved for future use. | | |
| 1000 | ConfigureApplication | Reserved for future use. | | |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 1100 | RestartApplication | Reserved for future use. | | |
| 1200 | EndProcessing | Trust Protection Foundation completes the certificate processing and, if configured, runs a Validation check on the certificate and application. | | x |
| 1400 | Revocation | Trust Protection Foundation submits a revocation request to the CA.<br>Certificate revocation is a certificate operation; it does not involve the application driver. | x | x |

## Permission requirements

When Trust Protection Foundation provisions certificates or conducts an onboard Validation on a supported iPlanet server, the user account that Trust Protection Foundation uses to authenticate with the module must have the following permissions:

- *Read* and *write* access to the Trust Database.
- *Read* and *write* access to the Temp Directory defined in the Device object.

  For more information on the Device object configuration, see "Creating a device object in the Policy Tree" on page 239.

**NOTE** In iPlanet Web Server, each server instance has its own certificate/key pair. The certificate/key pair is referred to as a Trust Database. The Trust Database is defined in the iPlanet Application object the Certificate Database. For more information, see "Creating an Oracle iPlanet application object" on page 526.

## Prerequisite configuration

Prior to using Trust Protection Foundation to manage certificates on iPlanet servers, you should complete the following:

1. Ensure the PK12util and Certutil utilities are installed.

   PK12util and Certutil are installed by default with a standard iPlanet install. You will need the path to both of these utilities when you configure the iPlanet Application object in Policy Tree. For more information on these utilities, refer to your iPlanet documentation.

2. If you plan to use SQLite, ensure that NSS 3.12 or newer is installed on the target server. For NSS 3.35 or newer, the Berkeley DB is not supported. Use the default SQLite database instead. For more information, see https://wiki.mozilla.org/NSS_SQLite-based_DB.

   This is because support for SQLite was added in NSS version 3.12. For more information, see "Certificate Database Type" on page 529.

   > **IMPORTANT** The CyberArk iPlanet driver does not support conversion of keystore databases to other formats. So, if you are provisioning to an existing database the Certificate Database Type must match the format of the existing database. If the setting does not match the existing format, provisioning will fail. For example, if you have an existing Berkeley database and you change the application object setting to SQLite, provisioning will fail.

3. Create a Trust Database.

   In iPlanet Web Server, each server instance has its own certificate/key pair. The certificate/key pair is referred to as a Trust Database. The Trust Database should be created only on your local machine. Virtual servers are covered by the Trust Database created for their server instance.

   In the Trust Database, you create and store the public and private keys in a file referred to as your key-pair file. The certificate is stored in the Trust Database after installation.

   For information on creating the Trust Database, refer to your iPlanet web server documentation.

4. Execute permissions to the Certutil and PK12util executables in the CLASS_PATH.

   Grant the following permissions to the user account that Trust Protection Foundation uses to authenticate with the keystore:

   - *Read* and *write* access to the certificate database (Trust Database) where Trust Protection Foundation installs the certificate, private key, and root certificate chain files. The certificate database is defined in the iPlanet Application object. For more information, see "Creating an

Oracle iPlanet application object" on page 526.

- Read and Write access to the Temp Directory defined in the Device object. For more information on the Device object configuration, see "Creating a device object in the Policy Tree" on page 239. The user account must be an owner of the keystore.

5. Activate SSL and set your encryption preferences for the iPlanet web server.

   For information on activating SSL and setting your encryption preferences, refer to your iPlanet web server documentation.

6. Open the SSH port.

   Trust Protection Foundation uses the Secure Shell (SSH) protocol to manage certificates on iPlanet web servers; therefore, Trust Protection Foundation must have access to the web server's SSH port. The default SSH port is port 22.

7. In the Policy Tree, create a server object for the server where the iPlanet web server is installed.

   For more information, see "Creating a device object in the Policy Tree" on page 239.

8. In the Policy Tree, create and configure an iPlanet application object.

   For more information on creating Application objects, see "Creating an application" on page 265. For details on the object's settings, see "Creating an Oracle iPlanet application object" on the next page.

9. In the Policy Tree, associate the iPlanet Application object with the certificates stored in the keystore.

   For more information, see Associating a certificate with an application from the certificate object.

## About iPlanet keystore backups

Whenever Trust Protection Foundation performs a certificate or private key operation, it creates a backup copy of the crypto file in the same directory where the original keystore resides.

If a backup operation fails, Trust Protection Foundation logs an event and halts processing. For example, if a backup operation fails for the JKS driver, the *JKS - Backup KeyStore Failed* event (ID 400E0020) is logged.

> **CAUTION** If a CyberArk driver cannot back up a crypto file, it will not attempt any further certificate and private key operations.

## Creating an Oracle iPlanet application object

To enable Trust Protection Foundation to manage PEM files installed on iPlanet servers, you must configure an iPlanet application object. This object provides the information Trust Protection Foundation needs to monitor, enroll, or provision PEM files on its associated iPlanet servers.

**DID YOU KNOW?** When you add an installation to a certificate, you'll have the option of defining (and editing) this object during that process, which means that you don't have to log in to Policy Tree as the following procedure describes. And because the settings are the same, you can use this topic for information about each setting.

For more information, see Creating a certificate installation in Aperture in the *Certificate Management Guide*.

**BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

### To create and configure an Oracle iPlanet application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **iPlanet**.
3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.
4. (Optional) In the **Device Certificate** box, click  to select and associate a certificate with the new application.

   **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

For detailed information on associating a certificate with an application, see Associating a certificate with an application object.

5. Under **General**, do the following:

   a. In the **Application Name** field, type a name for the new application.

   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.

   c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

      Default system notifications are sent to the contact identities. The default contact is the master administrator.

      > **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
      >
      > Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

   d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

      The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

   e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

      You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

      For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Under **Application Information**, do the following:

a. Click  next to **Application Credential** to browse for the credential object that you want to use to authenticate with the application.

> **DID YOU KNOW?** Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential might be a user name or private key credential; some drivers–such as F5, which is not SSH-based–can only use the user name credential for authentication.

> **NOTE** The user account you select must have *Read* and *Write* access to the Temporary, Private Key, and Certificate directories.
>
> If you need help with this step, see your system administrator.
>
> For more information, see Working with system credentials in the *Administration Guide*.

> **DID YOU KNOW?** The **Connection Method** is the protocol that Trust Protection Foundation uses to connect to the server and manage the certificates installed on that server. In an application object's settings, this field is typically read-only.

b. Click the **Connection Method** list, click the protocol to use–HTTPS or SSH–and then in the **Port** field, specify the associated port number.

c. (Optional) In the **Port** field, type the port that Trust Protection Foundation should use to communicate with the server where the application is installed.

Trust Protection Foundation uses the SSH protocol to communicate with the application server installed on Linux or Windows. The default SSH port assignment is port 22.

7. Complete the settings for the application object by referring to the following table:

| Field | Policy | Description |
|---|---|---|
| **iPlanet Settings** | | |
| Certificate Database Type | | Select the place where the iPlanet driver will provision keys and certificates:<br>◦ **Berkeley** (the default) . For NSS 3.35 or newer, the Berkeley DB is not supported. Use the default SQLite database instead. For more information, see https://wiki.mozilla.org/NSS_SQLite-based_DB.<br>◦ **SQLite**: Select SQLite if multiple processes share the same database and you want to be able to update the database while the server is running.<br>**NOTE** SQLite is supported only if NSS tools 3.12 are installed on the target server. See "Prerequisite configuration" on page 524. |
| Certificate Database Path |  | Path to the Certificate (Trust) Database.<br>For iPlanet 6.1, the default path to the Trust Database is as follows:<br>*server_root*/alias/<br>For iPlanet 7.0, the default path to the Trust Database is as follows:<br>*server_root*/https- *ConfigurationName*/config/<br>**NOTE** Refer to your iPlanet documentation for information on creating the Trust Database. |
| Certificate Database Credential |  | Password Trust Protection Foundation uses to authenticate with the Certificate Database.<br>**NOTE** Trust Protection Foundation does not include the database password on the command line when performing key management operations via certutil or pk12util. Instead, it causes the utility to prompt for the password. |

| Field | Policy | Description |
|---|---|---|
| Certificate Database Prefix |  | If you're using iPlanet 6.1, you must type a prefix here in order for the keystore to function properly. This string is then prepended to key3.db or cert8.db.<br>Type the prefix using the following syntax:<br>`https-`*VirtualServerName*`- `*HostName*`-`<br>Replace *VirtualServerName* and *HostName* with the actual values of your server.<br>Typically, iPlanet 7.0 doesn't require a prefix. |
| Create |  | If you want Trust Protection Foundation to create a new trust store automatically if one is not found, select **Yes**. |
| Replace Existing |  | If you selected Yes in the previous field (Create), you can then specify if you want Trust Protection Foundation to replace an existing trust store with a new one, if one is found.<br>**NOTE** If you want Replace Existing to always be enabled, you must enable it using a policy. This is because if you only set it here, it reverts to disabled (No) following the first successful provision. |
| **Utilities** | | |

| Field | Policy | Description |
|---|---|---|
| Certutil Path |  | Path to the Certutil keystore management utility. Trust Protection Foundation requires access to this utility to manage certificates and private keys in the Certificate Database.<br><br>**NOTE** The Certutil utility is installed by default with a standard iPlanet install. For more information about this utility, refer to your iPlanet documentation. |
| Pk12util Path |  | Path to the Pk12util keystore management utility. Trust Protection Foundation requires access to this utility to manage certificates and private keys in the Certificate Database.<br><br>**NOTE** The PK12util utility is installed by default with a standard iPlanet install. For more information about this utility, refer to your iPlanet documentation. |
| Certificate Alias |  | Alias that is assigned to the key/certificate in the keystore so it can be used by servers and applications accessing the Certificate Database.<br><br>**NOTE** This value must be unique within the Certificate Database. Since the Certificate Database only permits one instance of any given Common Name, the alias should not be changed unless the CN of the certificate is altered at the same time. When provisioning a certificate renewal to an existing Certificate Database, it is important that the alias remain the same as that of the certificate being renewed. |

8. (Optional) Under **File Ownership and Permissions**, select **Yes** on the **Set Owner and Permissions after Provisioning Files** drop-down–if you want to set specific permissions and ownership on files after they have been provisioned by Trust Protection Foundation–and then do the following:

   a. In the **Owner** field, type the user account name of the user who should have access to the provisioned files.

   **BEST PRACTICE** Who you assign as owners and approvers of your certificates is an important part of your PKI strategy. This is especially true because employees continue to pose the greatest threat to securing trust. Typically, this is because many employees fail to follow security best practices.

b. From the **Owner Permissions** list, select the level of permissions you want to grant to the owner (*Read*, or *Read and Write*).

c. In the **Group** field, type the group name to which the owner belongs.

d. From the **Group Permissions** list, select the level of permissions you want to grant to that group (*None*, *Read*, or *Read and Write*).

9. When you are finished, click **Save**.

## What's next?

After you've created an application object, here are other things you can do to manage the new application:

- On the application's **Settings** sub-tab:
  - Click  to push a certificate to its associated application.

    For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.
  - Click  **Reset** to stop processing the application and reset the status and stage.
  - Click  to reattempt installation of the certificate to its associated application, .
  - Click  **Validate Now** to validate the applications associated certificate.

    Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

    For more information, see "About certificate and application validation" on page 270.
- On the application object's **Validation** tab, you can configure validation settings for the application object.
- On an object's **General** tab:

- Click the **Log** sub-tab to view any events that are triggered by the template object.
- Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

## Palo Alto Networks configuration

Beginning with Trust Protection Foundation 20.4, the Palo Alto Networks application driver has been simplified to manage only device certificates for your Palo Alto appliance and no longer supports visibility functionality into encrypted network traffic.

If you want to install certificate private keys onto your Palo Alto Networks appliance using Trust Protection Foundation to decrypt your corporate traffic, then create a bulk provisioning job. For more information, see Installing (provisioning) lots of certificates and keys at one time in the Certificate Management Guide.

The foundation of the Palo Alto Networks enterprise security platform is a next-generation firewall capable of a wide range of security functions including network traffic decryption.

> **TIP** For version compatibility information, see "Supported integrations with Trust Protection Foundation" on page 18.

Having the visibility into network traffic that decryption provides is essential to threat detection and prevention but requires firewall administrators to regularly upload new digital certificates and encryption keys on the appliance. The sooner the firewall has those encryption assets after they have been activated on the respective hosting servers, the less time that network traffic is indecipherable.

The CyberArk Trust Protection Foundation streamlines the process of updating Palo Alto Networks firewalls. By supporting the direct provisioning of certificates and private keys to endpoint servers and the firewall, the period of time that the firewall is unable to decrypt traffic is minimized.

The Palo Alto Networks application driver also supports provisioning certificates to the Palo Alto Networks Firewall's trust store.

> **NOTE** For information about the trust store, see About creating and configuring trust stores in the *CyberArk Trust Protection Foundation Certificate Management Guide*.

This section provides the information you need to correctly configure a Palo Alto Networks application object.

This chapter contains the following topics:

## Palo Alto Networks Firewall Certificate Lifecycle

The following table outlines certificate lifecycle stages for certificates and private keys installed on a Palo Alto Networks firewall and identifies the management level associated with each stage.

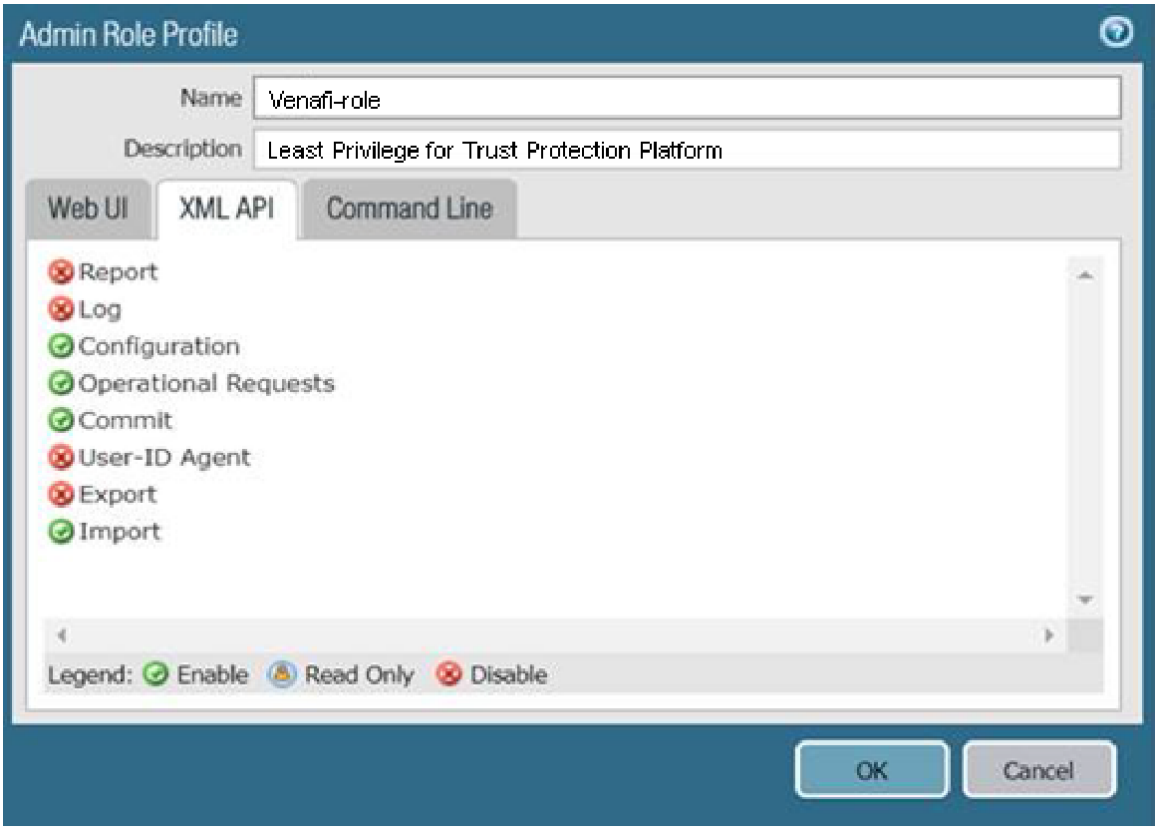
Lifecycle stages 800 through 1200 are handled by application drivers. However, not all stages are defined for every driver. For more information about lifecycle stages, see "About certificate lifecycle management" on page 42 in the *CyberArk Trust Protection Foundation Certificate Management Guide*.

For the Palo Alto Networks driver, stages 800 and 801 are defined in the following table.

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 800 | InstallChain | Optionally installs the root and intermediate certificate chain applicable to the certificate being provisioned.<br><br>If the Install Chain configuration option is set to Yes, then the chain is installed at Stage 800; otherwise, stage 800 is effectively skipped.<br><br>See the "Install Chain" on page 540 configuration option in "Creating a Palo Alto Networks Firewall application object" on the next page. | |  |
| 801 | InstallCertificate | Installs the certificate and private key. Optionally creates Decryption Policy Rules and binds certificates to them. | |  |

## Palo Alto Networks Firewall permission requirements

The user name credential that Trust Protection Foundation uses for authentication must be a Palo Alto Networks user who has been assigned to an Admin Role Profile with at least *Configuration*, *Operational Requests*, *Commit* and *Import* privileges to the XML API.




## Creating a Palo Alto Networks Firewall application object

To install an SSL certificate onto Palo Alto Networks Firewall, you must apply the required configuration settings described in this topic.

For version compatibility information, see "Supported integrations with Trust Protection Foundation" on page 18.

> **BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

### To create and configure a Palo Alto Networks Firewall application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **Palo Alto Networks**.

3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.

4. (Optional) In the **Device Certificate** box, click  to select and associate a certificate with the new application.

   > **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

   For detailed information on associating a certificate with an application, see Associating a certificate with an application object.

5. Under **General**, do the following:

   a. In the **Application Name** field, type a name for the new application.

   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.

   c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

      Default system notifications are sent to the contact identities. The default contact is the master administrator.

      > **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
      >
      > Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Under **Application Information**, do the following:

a. Click  next to **Application Credential** to browse for the credential object that you want to use to authenticate with the application.

> **DID YOU KNOW?** Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential might be a user name or private key credential; some drivers–such as F5, which is not SSH-based–can only use the user name credential for authentication.

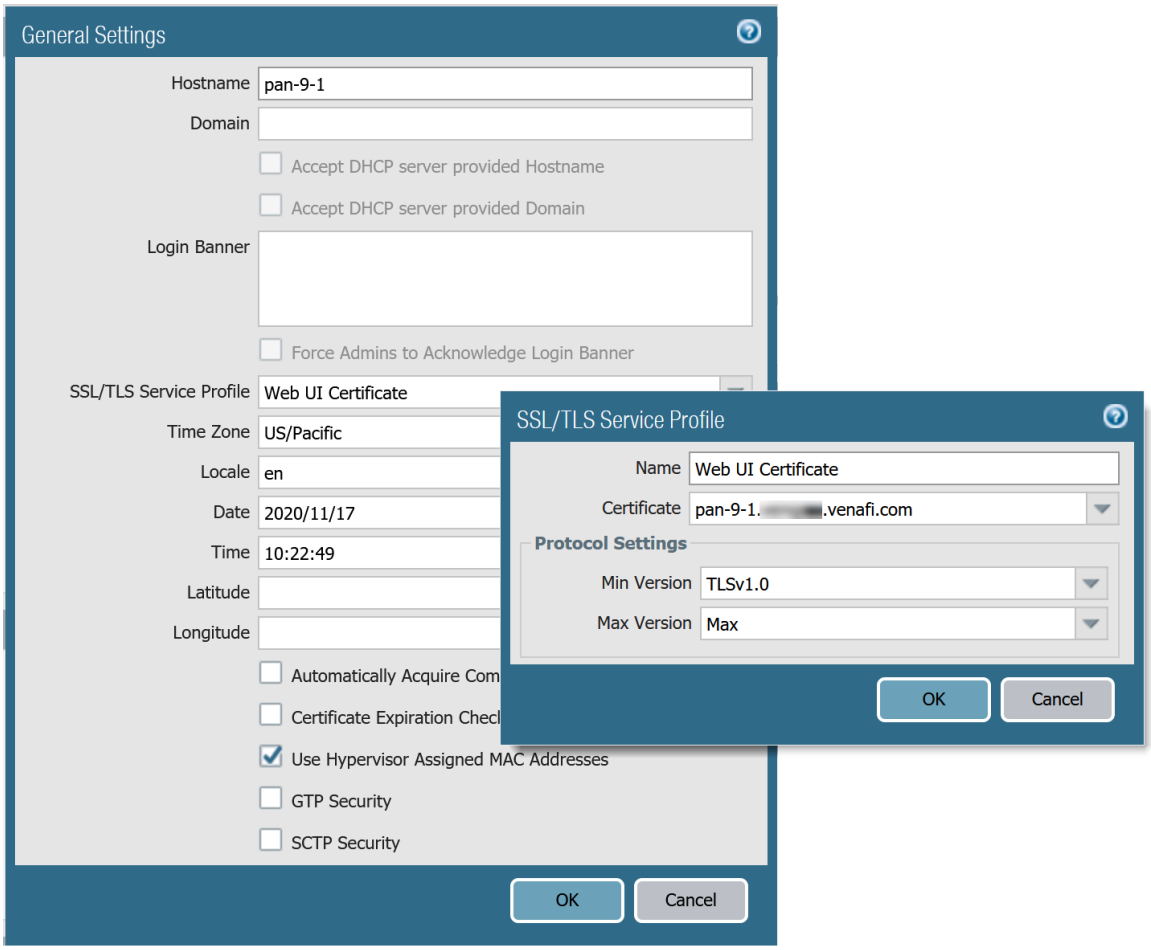
> **NOTE** The user account you select must have *Read* and *Write* access to the Temporary, Private Key, and Certificate directories.
>
> If you need help with this step, see your system administrator.
>
> For more information, see Working with system credentials in the *Administration Guide*.

b. (Optional) In the **Port** field, type the port that Trust Protection Foundation should use to communicate with the server where the application is installed.

Trust Protection Foundation uses the SSH protocol to communicate with the application server installed on Linux or Windows. The default SSH port assignment is port 22.

7. In the **Palo Alto Settings** box, refer to the following table to complete the settings for the application object:

| Field | Policy | Description |
|---|---|---|
| **Device Key Certificate** | | |
| Provision | | Controls whether Trust Protection Foundation installs only the certificate, or if it installs both the certificate and private key into the Device Certificates store on the Palo Alto Networks Firewall. |
| SSL/TLS Service Profile | | This is the configuration element that binds a certificate in the firewall's certificate store to the firewall's management interfaces (UI/API).<br><br>In the PAN console, you'll find it under **Device** > **Certificate Management** > **SSL/TLS Service Profile**. The binding to the management interface is under **Device** > **Setup** > **Management** tab. |



| Field | Policy | Description |
|---|---|---|
| Private Key Password |  | The password credential that is used to encrypt the private key when it is provisioned to the firewall. |
| Install Chain |  | This setting controls whether or not the root and intermediate certificate chain is installed at the same time the certificate and private key are provisioned.<br><br>When not specified, the default behavior is not to install the chain. |
| Replace Certificate |  | Deletes an existing certificate to allow for the new certificate with the same Subject distinguished name to be provisioned.<br><br>The Palo Alto Networks Firewall allows only one instance of a Subject DN to exist in its Device Certificates store. |
| **Decryption** | | |
| Decryption Policy Rule Name | | The name of the Decryption Policy Rule on the Palo Alto Networks Firewall to which the certificate should be bound after it has been successfully provisioned. |
| Create Decryption Policy Rule |  | When this setting is set to Yes, the Trust Protection Foundation creates the specified Decryption Policy Rule if it does not yet exist. |
| Decryption Profile Name |  | The name of the Decryption Profile to assign to the Decryption Policy Rule when it is created by the Trust Protection Foundation. If the Decryption Profile does not exist, then it is created. |
| Destination Address(es) | | IP addresses, IP ranges, CIDR subnets, and/or FQDNs that are assigned as Destination Addresses for the Decryption Policy Rule when it is created by the Trust Protection Foundation. |
| **Device Config Lock** | | |
| Lock Config |  | This setting controls whether Trust Protection Foundation should lock the Palo Alto Networks Firewall configuration while it is actively provisioning the certificate. Locking the configuration prevents the driver from committing changes that it did not make.<br><br>If the configuration is not locked, the driver may commit any changes that were already pending on the firewall at the time of provisioning. |

1. When you are finished, click **Save**.

### What's next?

After you've created an application object, here are other things you can do to manage the new application:

- On the application's **Settings** sub-tab:
  - Click  to push a certificate to its associated application.

    For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.
  - Click  **Reset** to stop processing the application and reset the status and stage.
  - Click  to reattempt installation of the certificate to its associated application, .
  - Click  **Validate Now** to validate the applications associated certificate.

    Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

    For more information, see "About certificate and application validation" on page 270.
- On the application object's **Validation** tab, you can configure validation settings for the application object.
- On an object's **General** tab:
  - Click the **Log** sub-tab to view any events that are triggered by the template object.
  - Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

## PEM configuration

Privacy-enhanced Electronic Mail (PEM) is a standard originally used for distributing X.509 certificates in plain text via e-mail using Base64 encoding of DER certificates. Originally used for encrypting the contents of e-mail, the PEM format is now used by many applications. The CyberArk Trust Protection Foundation™ PEM driver supports management of PEM files.

In Trust Protection Foundation, the PEM application object is used to provision PEM certificates and can provision the end-entity, chain, and as of Trust Protection Foundation version 18.3, private key to a single file (rather than two or three separate files).

This section provides the information you need to manage PEM files.

> **NOTE** For information about the trust store, see About creating and configuring trust stores in the *CyberArk Trust Protection Foundation Certificate Management Guide*.

This chapter contains the following topics:

## PEM Certificate Lifecycle

The following table outlines the stages of the certificate lifecycle for PEM certificates and the management level associated with each stage.

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| Stages 0-700 are performed by the PEM Application driver only if remote key generation is enabled. If the private key and CSR are locally generated on the Trust Protection Foundation server, stages 0-700 are performed by the X509Certificate Application driver.<br><br>The private key and CSR are remotely generated on the certificate's consumer application(s) if the **Generate Key/CSR on Application** option is enabled in the Certificate object. The **Generate Key/CSR on Application** option is not supported in PEM applications on Windows server environments. | | | | |
| 0 | StartProcessing | No processing. | | |
| 100 | CheckStore | No processing. | | |
| 200 | CreateConfigureStore | No processing. | | |
| 300 | CreateKey | Trust Protection Foundation creates the private key. | x | x |
| 400 | CreateCSR | Trust Protection Foundation creates the Certificate Signing Request (CSR). | x | x |
| 500 | PostCSR | Trust Protection Foundation submits the CSR to the Certificate Authority (CA). | x | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| | | If you post a manual CSR, this is the first stage of the certificate lifecycle. | | |
| 600 | ApproveRequest | Trust Protection Foundation approves the certificate renewal at the CA. | x | x |
| 700 | RetrieveCertificate | Trust Protection Foundation retrieves the certificate from the CA. | x | x |
| 800 | InstallCertificateChain | Trust Protection Foundation installs the root certificate chain file if it is available.<br><br>If there is an existing chain file, Trust Protection Foundation will install the new certificate chain file only if the **Overwrite Existing Chain** option is selected in the PEM Application object.<br><br>Root certificates can be discovered and brought under management, or they can be manually imported. For more information, see Managing Root Certificates. | | x |
| 801 | InstallPrivateKey | Trust Protection Foundation installs the private key if the private key was not created on the host. | | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 802 | InstallCertificate | Trust Protection Foundation installs the certificate in the keystore or Directory designated in the PEM driver configuration. | | x |
| 900 | CheckConfiguration | Trust Protection Foundation verifies the configuration after the certificate is installed. | | |
| 1000 | ConfigureApplication | Trust Protection Foundation configures the application to use the installed certificate, if needed. | | |
| 1100 | RestartApplication | Trust Protection Foundation restarts the application after the certificate is installed and configured, if needed. | | |
| 1200 | EndProcessing | Trust Protection Foundation completes the certificate processing and, if configured, runs a Validation check on the certificate and application. | | x |
| 1400 | Revocation | Trust Protection Foundation submits a revocation request to the CA.<br>Certificate revocation is a certificate operation; it does not involve the application driver. | x | x |

## Permission Requirements

When Trust Protection Foundation provisions certificates or conducts an onboard Validation on an PEM application server, the user account that Trust Protection Foundation uses to authenticate with the server must have the following permissions:

- Read and Write access to the directories where Trust Protection Foundation installs the certificate, private key, and root certificate chain files. These directories are defined in the PEM Application object. For more information, see "Creating a PEM application object" on the facing page.
- Read and Write access to the Temp Directory defined in the Device object. For more information on the Device object configuration, see "Managing device objects" on page 238.

## PEM prerequisite configuration

To enable Trust Protection Foundation to provision PEM certificates, you must complete the following high-level tasks:

1. Create the Directory where you want Trust Protection Foundation to create the PEM file.
2. Grant the user account that Trust Protection Foundation uses to authenticate to the host server Read and Write access to the PEM file.

   The location of the PEM file is defined in the PEM Application object. For more information, see "Creating a PEM application object" on the facing page.
3. Open the SSH port.

   Trust Protection Foundation uses the Secure Copy (SCP) protocol to copy the PEM file to the target Directory. Trust Protection Foundation must have access to the SCP port. The default SCP port is port 22.
4. In the Policy Tree, create a Device object for the server where the PEM file is located.

   For more information, see "Managing device objects" on page 238.
5. In the Policy Tree, create and configure an Application object for the PEM file.

   For more information on creating Application objects, see "Managing application objects" on page 259. For details on the object's settings, see "Creating a PEM application object" on the facing page.

6. In the Policy Tree, associate the PEM Application object with the certificate you want to write to the PEM file.

   For more information, see Associating Certificates with Applications.

## About PEM file backups

Whenever Trust Protection Foundation performs a certificate or private key operation, it creates a backup copy of the crypto file in the same directory where the original keystore resides.

If a backup operation fails, Trust Protection Foundation logs an event and halts processing. For example, if a backup operation fails for the JKS driver, the *JKS - Backup KeyStore Failed* event (ID 400E0020) is logged.

**CAUTION** If a CyberArk driver cannot back up a crypto file, it will not attempt any further certificate and private key operations.

## Creating a PEM application object

To enable Trust Protection Foundation to manage PEM certificates, you must configure the PEM application object. This object provides the information Trust Protection Foundation needs to monitor, enroll, or provision PEM files on its associated servers, and can even provision the end-entity, chain and– as of Trust Protection Foundation version 18.3– private key to a single file (rather than two or three separate files).

**BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

**DID YOU KNOW?** When you add an installation to a certificate, you'll have the option of defining (and editing) this object during that process, which means that you don't have to log in to Policy Tree as the following procedure describes. And because the settings are the same, you can use this topic for information about each setting.

For more information, see Creating a certificate installation in Aperture in the *Certificate Management Guide*.

### To create and configure a PEM application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **PEM**.
3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.
4. (Optional) In the **Device Certificate** box, click ... to select and associate a certificate with the new application.

   > **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

   For detailed information on associating a certificate with an application, see Associating a certificate with an application object.
5. Under **General**, do the following:
   a. In the **Application Name** field, type a name for the new application.
   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.
   c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

      Default system notifications are sent to the contact identities. The default contact is the master administrator.

> **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
>
> Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

   The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

   You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

   For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Under **Application Information**, do the following:

   a. Click  next to **Application Credential** to browse for the credential object that you want to use to authenticate with the application.

   > **DID YOU KNOW?** Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential might be a user name or private key credential; some drivers–such as F5, which is not SSH-based–can only use the user name credential for authentication.

   > **NOTE** The user account you select must have *Read* and *Write* access to the Temporary, Private Key, and Certificate directories.
   >
   > If you need help with this step, see your system administrator.
   >
   > For more information, see Working with system credentials in the *Administration Guide*.

> **DID YOU KNOW?** The **Connection Method** is the protocol that Trust Protection Foundation uses to connect to the server and manage the certificates installed on that server. In an application object's settings, this field is typically read-only.

b. Click the **Connection Method** list, click the protocol to use–HTTPS or SSH–and then in the **Port** field, specify the associated port number.

c. (Optional) In the **Port** field, type the port that Trust Protection Foundation should use to communicate with the server where the application is installed.

   Trust Protection Foundation uses the SSH protocol to communicate with the application server installed on Linux or Windows. The default SSH port assignment is port 22.

7. Complete the PEM application settings by referring to the following table:

| Field | Policy | Description |
|---|---|---|
| **PEM Settings** | | The following are server-specific certificate settings. They are referenced only when you associate a certificate with the current PEM Application object. |
| Private Key File |  | The path and filename where Directory Certificate Manager - Self-Hosted installs the private key.<br><br>This setting must match the PEM application's private key file configuration. For more information, refer to your PEM documentation. |
| Private Key Credential |  | The credential required to access the private key file for certificate renewal.<br><br>**To select a private key password credential**<br>a. Click  to open the **Credential Selector** dialog.<br><br>b. Select the credential required to access the private key file for certificate renewal, and then click **Select**.<br><br>For more information, see Managing system credentials in the *CyberArk Trust Protection Foundation Administration Guide*.<br><br>Trust Protection Foundation does not include the private key password on the command line when performing key management operations. Instead, it provides the password when prompted. |
| Private Key PBE Algorithm | | The Password-Based Cryptography Specification Version 2.0 (PBES2) algorithm.<br><br>Each algorithm type has a corresponding security/compatibility value. Generally they are inversely related due to their adoption by software applications. |
| Private Key Syntax |  | The format specification for the private key. |

| Field | Policy | Description |
|---|---|---|
| | | Existing PEM objects default to OpenSSL (PKCS#1). When creating a new object, the default is PKCS#8.<br><br>OpenSSL is only allowed when the Private Key PBE Algorithm is set to SHA1/3DES Insecure but better system compatibility or SHA256/AES256 high security but low system compatibility.<br><br>**NOTE** ECC remote generation support requires an OpenSSL version 1.0.2 or higher installed on the target device |
| Certificate File |  | The path and filename on the PEM application server where Trust Protection Foundation installs the certificate.<br><br>This setting must match the PEM application's certificate file configuration. For more information, refer to your PEM documentation.<br><br>**DID YOU KNOW?** If you want to provision the end-entity, chain and private key to a single file (rather than two or three separate files), then specify the same path and filename for the Certificate File and Certificate Chain File settings, and/or Private Key File settings on the application object. |
| Certificate Chain File |  | The path and filename on the PEM application server where Trust Protection Foundation writes root certificates.<br><br>This setting must match the PEM application's certificate chain file configuration. For more information, refer to your PEM documentation. |
| Overwrite Existing Chain |  | Overwrites the existing certificate chain file when Trust Protection Foundation installs a new certificate and private key. |

8. (Optional) Under **File Ownership and Permissions**, select **Yes** on the **Set Owner and Permissions after Provisioning Files** drop-down–if you want to set specific permissions and ownership on files after they have been provisioned by Trust Protection Foundation–and then do the following:

    a. In the **Owner** field, type the user account name of the user who should have access to the provisioned files.

> **BEST PRACTICE** Who you assign as owners and approvers of your certificates is an important part of your PKI strategy. This is especially true because employees continue to pose the greatest threat to securing trust. Typically, this is because many employees fail to follow security best practices.

b. From the **Owner Permissions** list, select the level of permissions you want to grant to the owner (*Read*, or *Read and Write*).

c. In the **Group** field, type the group name to which the owner belongs.

d. From the **Group Permissions** list, select the level of permissions you want to grant to that group (*None*, *Read*, or *Read and Write*).

9. When you are finished, click **Save**.

## What's next?

After you've created an application object, here are other things you can do to manage the new application:

- On the application's **Settings** sub-tab:
  - Click  to push a certificate to its associated application.

    For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.
  - Click  **Reset** to stop processing the application and reset the status and stage.
  - Click  to reattempt installation of the certificate to its associated application, .
  - Click  **Validate Now** to validate the applications associated certificate.

    Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

    For more information, see "About certificate and application validation" on page 270.
- On the application object's **Validation** tab, you can configure validation settings for the application object.
- On an object's **General** tab:

- Click the **Log** sub-tab to view any events that are triggered by the template object.
- Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

# PKCS#12 configuration

Public Key Cryptography Standard #12 (PKCS#12) is the Personal Information Exchange Syntax Standard that defines the file format used to store private keys with accompanying public key certificates and protect the files with a password-based symmetric key.

In CyberArk Trust Protection Foundation™, the PKCS#12 application object is used to provision PKCS#12 certificates to a PFX file. This section provides the information you need to manage PKCS#12 certificates.

Trust Protection Foundation's PKCS#12 driver supports only the **Linux Shell and SFTP** security setting in the latest version of COPSSH. It does not support the **Linux Shell**, **Windows Shell**, and **SFTP Only** security settings.

The PKCS#12 application driver also supports provisioning certificates to a PKCS#12 trust store.

> **NOTE** For information about the trust store, see About creating and configuring trust stores in the *CyberArk Trust Protection Foundation Certificate Management Guide*.

This chapter contains the following topics:

## PKCS#12 Certificate Lifecycle

The following table outlines the stages of the certificate lifecycle for PKCS#12 certificates and the management level associated with each stage.

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 0 | StartProcessing | No processing. | | |
| 100 | CheckStore | No processing. | | |
| 200 | CreateConfigureStore | No processing. | | |
| 300 | CreateKey | No processing. | | |
| 400 | CreateCSR | No processing. | | |
| 500 | PostCSR | No processing. | | |
| 600 | ApproveRequest | No processing. | | |
| 700 | RetrieveCertificate | No processing. | | |
| 800 | InstallCertificate | Trust Protection Foundation installs the PKCS#12 file to the target Directory.<br><br>If the Bundle Chain option is not selected, Trust Protection Foundation installs two files–the certificate and private key file and the root certificate chain file. | | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 900 | CheckConfiguration | Reserved for future use. | | |
| 1000 | ConfigureApplication | Reserved for future use. | | |
| 1100 | RestartApplication | Reserved for future use. | | |
| 1200 | EndProcessing | Trust Protection Foundation completes the certificate processing and, if configured, runs a Validation check on the certificate and application. | | x |
| 1400 | Revocation | Trust Protection Foundation submits a revocation request to the CA.<br><br>**NOTE** Certificate revocation is a certificate operation; it does not involve the application driver. | x | x |

## PKCS#12 Permission Requirements

When Trust Protection Foundation provisions certificates or conducts an onboard validation of a PKCS#12 file, the user account that Trust Protection Foundation uses to authenticate with the host server (that is, the server where the PKCS#12 file is located) must have *read* and *write* access to the PKCS#12 file.

The location of the PKCS#12 file is defined in the PKCS#12 Application object. For more information, see "Creating a PKCS#12 application object" on the facing page.

## About PKCS#12 keystore backups

Whenever Trust Protection Foundation performs a certificate or private key operation, it creates a backup copy of the crypto file in the same directory where the original keystore resides.

If a backup operation fails, Trust Protection Foundation logs an event and halts processing. For example, if a backup operation fails for the JKS driver, the *JKS - Backup KeyStore Failed* event (ID 400E0020) is logged.

**CAUTION** If a CyberArk driver cannot back up a crypto file, it will not attempt any further certificate and private key operations.

## Creating a PKCS#12 application object

To enable Trust Protection Foundation to provision PKCS#12 certificates to a PFX file, you must configure the PKCS#12 application object. This object provides the information Trust Protection Foundation needs to monitor, enroll, or provision PKCS#12 certificates.

**BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

**DID YOU KNOW?** When you add an installation to a certificate, you'll have the option of defining (and editing) this object during that process, which means that you don't have to log in to Policy Tree as the following procedure describes. And because the settings are the same, you can use this topic for information about each setting.

For more information, see Creating a certificate installation in Aperture in the *Certificate Management Guide*.

### To create and configure a PKCS#12 application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **PKCS#12**.
3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.

4. (Optional) In the **Device Certificate** box, click ... to select and associate a certificate with the new application.

   > **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

   For detailed information on associating a certificate with an application, see Associating a certificate with an application object.

5. Under **General**, do the following:

   a. In the **Application Name** field, type a name for the new application.

   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.

   c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

      Default system notifications are sent to the contact identities. The default contact is the master administrator.

      > **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
      >
      > Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

   d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

      The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Under **Application Information**, do the following:

a. Click  next to **Application Credential** to browse for the credential object that you want to use to authenticate with the application.

> **DID YOU KNOW?** Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential might be a user name or private key credential; some drivers—such as F5, which is not SSH-based—can only use the user name credential for authentication.

> **NOTE** The user account you select must have *Read* and *Write* access to the Temporary, Private Key, and Certificate directories.
>
> If you need help with this step, see your system administrator.
>
> For more information, see Working with system credentials in the *Administration Guide*.

> **DID YOU KNOW?** The **Connection Method** is the protocol that Trust Protection Foundation uses to connect to the server and manage the certificates installed on that server. In an application object's settings, this field is typically read-only.

b. (Optional) In the **Port** field, type the port that Trust Protection Foundation should use to communicate with the server where the application is installed.

Trust Protection Foundation uses the SSH protocol to communicate with the application server installed on Linux or Windows. The default SSH port assignment is port 22.

Complete the settings for the application object by referring to the following table:

| Field | Policy | Description |
|---|---|---|
| **PKCS#12 Settings** | | |
| PKCS#12 File |  | Complete path and filename of the PKCS#12 PFX file. The PFX file contains the private key and signed client certificate.<br><br>The user account that Trust Protection Foundation uses to authenticate to the host server must have Read and Write access to the PKCS#12 file. |
| Private Key Credential |  | This is the password credential that Trust Protection Foundation uses to encrypt both the certificate's private key and the entire PKCS#12 keystore.<br><br>This field is not required.<br><br>Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. In this case, the stored credential is a password.<br><br>**To select a private key password credential**<br><br>1. Click  to open the **Credential Selector** dialog.<br><br>2. Select the credential required to access the private key file for certificate renewal, and then click **Select**.<br><br>For more information, see Managing system credentials in the *CyberArk Trust Protection Foundation Administration Guide*.<br><br>**DID YOU KNOW?** Trust Protection Foundation does not include the private key password on the command line when performing key management operations. Instead, it provides the password when prompted. |
| Friendly Name | | Label assigned to the key/certificate in the keystore so it can be used by the server or application. |

| Field | Policy | Description |
|---|---|---|
| | | This value must be unique within the keystore. If you provision to an existing keystore using a friendly name that already exists in that keystore, and you have Replace Existing set to *No* (the default setting) and Reuse Friendly Name set to *No*, you'll receive an error message.<br><br>**DID YOU KNOW?** Depending on the trust store, this setting goes by different names. For example, GSK refers to this setting as the *Certificate Label*, JKS refers to it as the *Certificate Alias*, and CAPI and PKCS#12 refer to it as the *Friendly Name*.<br><br>**NOTE** When certificates are discovered by the Server Agent, the agent captures and sends the certificate label to Trust Protection Foundation using the existing friendly name JSON object property. Trust Protection Foundation then sets the certificate label value when it creates the new application object. |
| Certificate Chain Location |  | Determines where the certificate chain is located.<br><br>▪ **Store in PKCS#12 File** stores all the certificate's root and intermediate root certificates in the PKCS#12 file with the certificate.<br><br>▪ **Store in Certificate Chain File** stores the certificate's root and intermediate root certificates in a separate certificate chain file. When you select this option, the Certificate Chain File option becomes enabled so you can define the path and filename for the Certificate Chain File. |
| Certificate Chain File |  | Path and filename where Trust Protection Foundation writes the certificate's associated root and intermediate root certificates. This option is only available if you select **Store in Certificate Chain File** for the Certificate Chain Location. |

| Field | Policy | Description |
|---|---|---|
| | | The user account that Trust Protection Foundation uses to authenticate to the host server must have Read and Write access to the Certificate Chain Directory. |
| Create | | Creates a new PKCS#12 file if one does not already exist. The default is Yes. |
| Replace Existing | | When enabled (set to **Yes**), Trust Protection Foundation creates a backup of the original keystore and then creates a new one.<br><br>This option is available only when the **Create** option is selected, and you must select **Yes** if you want to enable it (the default setting is No).<br><br>**TIP** This option is reset to the default setting (No) automatically after Trust Protection Foundation has replaced the keystore. |
| Reuse Friendly Name | | Reuses the Friendly Name that is assigned to the key/certificate in the PKCS#12 file when the certificate is renewed.<br><br>This option keeps the existing certificate available during the renewal process and simplifies management of the applications that use the key or certificate referenced by the label. The default is Yes. |

7. (Optional) Under **File Ownership and Permissions**, select **Yes** on the **Set Owner and Permissions after Provisioning Files** drop-down–if you want to set specific permissions and ownership on files after they have been provisioned by Trust Protection Foundation–and then do the following:

   a. In the **Owner** field, type the user account name of the user who should have access to the provisioned files.

   **BEST PRACTICE** Who you assign as owners and approvers of your certificates is an important part of your PKI strategy. This is especially true because employees continue to pose the greatest threat to securing trust. Typically, this is because many employees fail to follow security best practices.

b. From the **Owner Permissions** list, select the level of permissions you want to grant to the owner (*Read*, or *Read and Write*).

c. In the **Group** field, type the group name to which the owner belongs.

d. From the **Group Permissions** list, select the level of permissions you want to grant to that group (*None*, *Read*, or *Read and Write*).

8. When you are finished, click **Save**.

## What's next?

After you've created an application object, here are other things you can do to manage the new application:

- On the application's **Settings** sub-tab:
  - Click  to push a certificate to its associated application.

    For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.
  - Click  **Reset** to stop processing the application and reset the status and stage.
  - Click  to reattempt installation of the certificate to its associated application, .
  - Click  **Validate Now** to validate the applications associated certificate.

    Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

    For more information, see "About certificate and application validation" on page 270.
- On the application object's **Validation** tab, you can configure validation settings for the application object.
- On an object's **General** tab:

- Click the **Log** sub-tab to view any events that are triggered by the template object.
- Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

# Riverbed SteelHead Configuration

In many enterprise environments, applications are distributed geographically. Distributing applications can help to reduce costs, provide higher availability, and can be useful when knowledgeable IT resources are located in geographies outside of an application's customer base. Unfortunately, remote applications often suffer from poor performance caused by high network latency and low throughput.

Riverbed SteelHead's wide area network (WAN) optimization products strive to solve these problems by compressing and de-duplicating network traffic. However, for encrypted traffic, Riverbed SteelHead's products can only be effective when data can be effectively decrypted and inspected.

The CyberArk Trust Protection Foundation streamlines the process of updating Riverbed SteelHead appliances to ensure that encrypted traffic is processed along with non-encyrpted traffic.

This section provides the information you need to correctly configure a Riverbed SteelHead application object.

This chapter contains the following topics:

## Riverbed SteelHead Certificate Lifecycle

The following table outlines certificate lifecycle stages for certificates and private keys installed on a Riverbed SteelHead WAN optimization appliance and identifies the management level associated with each stage.

> Lifecycle stages 800 through 1200 are handled by application drivers. However, not all stages are defined for every driver. For more information about lifecycle stages, see "About certificate lifecycle management" on page 42 in the *CyberArk Trust Protection Foundation Certificate Management Guide*.

For the Riverbed SteelHead driver, stages 800 and 801 are defined in the following table.

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 800 | InstallCertificate | Installs the certificate and private key. | |  |
| 801 | InstallChain | Optionally installs the root and intermediate certificate chain applicable to the certificate being provisioned.<br><br>If the Install Chain configuration option is set to Yes, then the chain is installed at stage 801; otherwise, stage 801 is effectively skipped.<br><br>See the "Install Chain" on page 540 configuration option in "Creating a Palo Alto Networks Firewall application object" on page 536. | |  |

## Riverbed SteelHead permission requirements

The username credential used by CyberArk Trust Protection Foundation™ for authentication must be an administrative user with privileges that allow Trust Protection Foundation to make configuration changes from an SSH command line interface (CLI) of the Riverbed SteelHead appliance.

## Creating a Riverbed SteelHeadapplication object

To install an SSL certificate onto a Riverbed SteelHead appliance, you must apply the required configuration settings described in this topic.

> **BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

### To create and configure a Riverbed SteelHead application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **Riverbed SteelHead**.
3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.
4. (Optional) In the **Device Certificate** box, click  to select and associate a certificate with the new application.

   > **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

   For detailed information on associating a certificate with an application, see Associating a certificate with an application object.
5. Under **General**, do the following:
   a. In the **Application Name** field, type a name for the new application.
   b. (Optional) In the **Description** field, type a description for the purpose of the application.

A strong description can help to provide context for other administrators who might need to manage the new application.

c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

Default system notifications are sent to the contact identities. The default contact is the master administrator.

> **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
>
> Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Complete the settings for the application object by referring to the following table:

| Field | Policy | Description |
|---|---|---|
| **Riverbed SteelHead Settings** | | |
| Certificate Type |  | Specifies in which of the three places that exist on the Riverbed SteelHead appliance for certificates that the Trust Protection Foundation will target when provisioning the certificate to the device.<br>◦ SSL Main<br>◦ Secure Peering<br>◦ Web |
| Replace Existing |  | If set to Yes, then when a certificate is encountered on the target device that has the same common name as the certificate being provisioned, the current certificate is deleted and replaced by the newly provisioned certificate.<br>If set to No when a certificate is encountered on the target device that has the same common name as the certificate being provisioned, processing will halt in error.<br>If you want to keep the old certificate, you should just rename it on the Riverbed SteelHead device and then retry from Trust Protection Foundation. |
| Install Chain |  | This setting controls whether or not the root and intermediate certificate chain is installed at the same time that the certificate and private key are provisioned.<br>When not specified, the default behavior is *not* to install the chain. |

7. When you are finished, click **Save**.

## What's next?

After you've created an application object, here are other things you can do to manage the new application:

- On the application's **Settings** sub-tab:
  - Click  to push a certificate to its associated application.

    For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.
  - Click  **Reset** to stop processing the application and reset the status and stage.
  - Click  to reattempt installation of the certificate to its associated application, .
  - Click  **Validate Now** to validate the applications associated certificate.

    Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

    For more information, see "About certificate and application validation" on page 270.
- On the application object's **Validation** tab, you can configure validation settings for the application object.
- On an object's **General** tab:
  - Click the **Log** sub-tab to view any events that are triggered by the template object.
  - Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

## Tealeaf PCA configuration

Tealeaf CX is a customer experience management product from Tealeaf Technology. This product performs non-intrusive capture and analysis of web traffic in order for businesses to better understand how customers experience their website.

In order to perform this type of analysis against websites encrypted using SSL, Tealeaf provides a component called the “Passive Capture Application” or PCA. The PCA server is connected to the network infrastructure via a collection device, network tap, or switch spanning port. SSL traffic is then decrypted using the same private key that the web server uses, stored on the PCA server in Tealeaf’s proprietary PTL format.

> **NOTE** The Tealeaf PCA provisioning driver supports only the storage of PTL private keys on a TeaLeaf PCA server and does not support the use of an HSM.

This section provides the information required to provision web server certificates' private keys to Tealeaf PCA servers.

This chapter contains the following topics:

## Tealeaf PCA Certificate Lifecycle

The following table outlines the lifecycle stages for private keys installed on Tealeaf PCA devices and the management level associated with each stage.

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| Stages 0-700 are performed by the X.509 Certificate Application driver on the Trust Protection Foundation server. | | | | |
| 0 | StartProcessing | No processing. | | |
| 100 | CheckStore | No processing. | | |
| 200 | CreateConfigureStore | No processing. | | |
| 300 | CreateKey | No processing. | | |
| 400 | CreateCSR | No processing. | | |
| 500 | PostCSR | No processing. | | |
| 600 | ApproveRequest | No processing. | | |
| 700 | RetrieveCertificate | No processing. | | |
| 800 | InstallPrivateKey | Trust Protection Foundation places a private key in PTL format on the host. If a PTL file with the same name exists, Trust Protection Foundation overwrites it. | | x |

## Permission Requirements

When Trust Protection Foundation provisions private keys or conducts an onboard Validation on a Tealeaf PCA server, the user account that Trust Protection Foundation uses to authenticate with the device must have the following permissions:

- Sufficient permissions to access the private key stored in the associated certificate.
- *Read* and *write* access to the Passive Capture Setup Path on the device. For more information, see "Creating a Tealeaf PCA application object" on page 576 and "Creating a device object in the Policy Tree" on page 239.

## Tealeaf PCA prerequisite configuration

Trust Protection Foundation automates the process of installing and maintaining your web servers' private keys on the Tealeaf PCA server.

To allow Trust Protection Foundation to provision your web server certificate's private keys on Tealeaf PCA servers, you must complete the following high-level tasks:

1. Open the SSH port to the Tealeaf PCA server.

   Trust Protection Foundation uses the Secure Shell (SSH) protocol to manage private keys; therefore, Trust Protection Foundation must have access to the device's SSH port. The default SSH port is port 22.

2. (Recommended) Create a Policy object for the Tealeaf PCA server and all its associated web servers and certificates.

   The Tealeaf PCA server and its associated web servers may have a one-to-many configuration. Consequently, it is easier to group all objects associated with the Tealeaf PCA server configuration under a single Policy object.

3. In Policy Tree, create a Device object for the machine that hosts the Tealeaf PCA server.

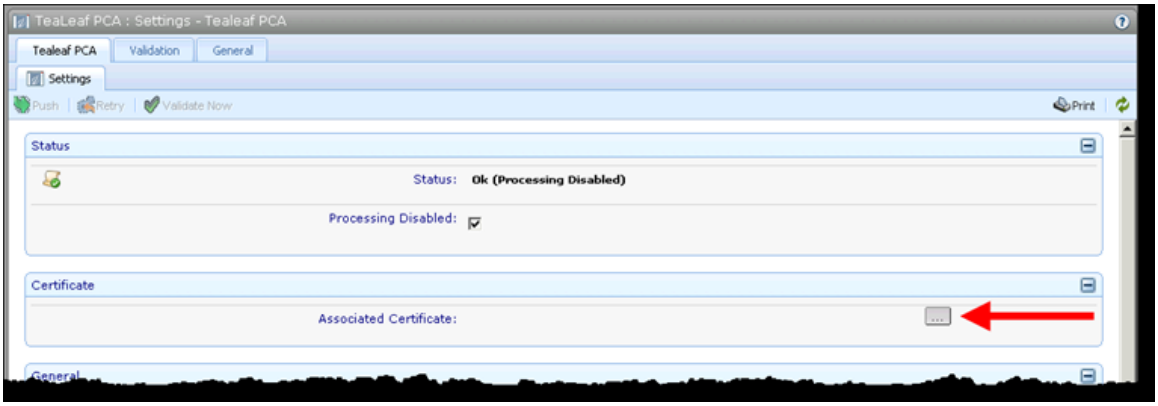
   For more information, see "Creating a device object in the Policy Tree" on page 239.

4. In Policy Tree, create and configure an Application object for the SSL Certificate private key to be exported to the Tealeaf PCA server.

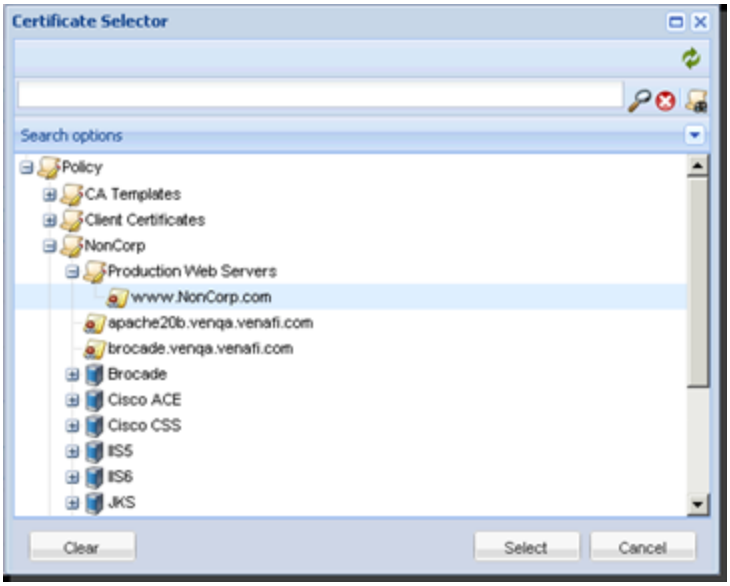
Unlike most applications managed by Trust Protection Foundation, the Tealeaf PCA application references the certificate being hosted by a web server or group of web servers in order to decrypt traffic destined for that server. Each application object references a single certificate, so if you have multiple SSL certificate private keys that Tealeaf CX needs for decryption purposes, you will need multiple application objects.

For more information on creating Application objects, see "Creating an application" on page 265. For details on the object's settings, see "Creating a Tealeaf PCA application object" on page 576.

5. In the Tealeaf PCA Application object's Associated Certificate field, click the **Browse** button.



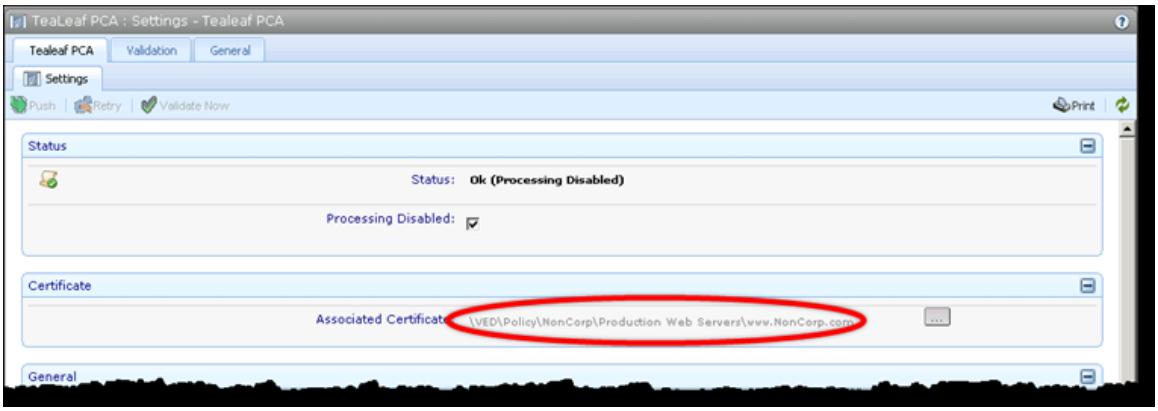

6. In the Certificate Selector dialog box, select the certificate installed on a web server that Tealeaf PCA is monitoring.



You must have either Write or both Associate and View permissions to the Certificate object to select it in the Certificate Selector dialog

7. Click **Select**.

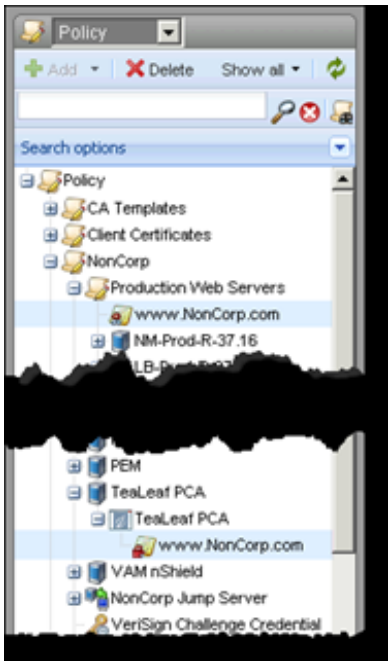
   The Certificate object is associated with the current application.




8. (Optional) Repeat steps 4-"Click Select." above for every certificate installed on a web server that the Tealeaf PCA server is monitoring.

   If enabled, an Alias object for the web server certificate associated with the Tealeaf PCA server appears under the Tealeaf PCA Application object.

   The certificate Alias objects also appear under the web server Application objects where the certificate is installed.



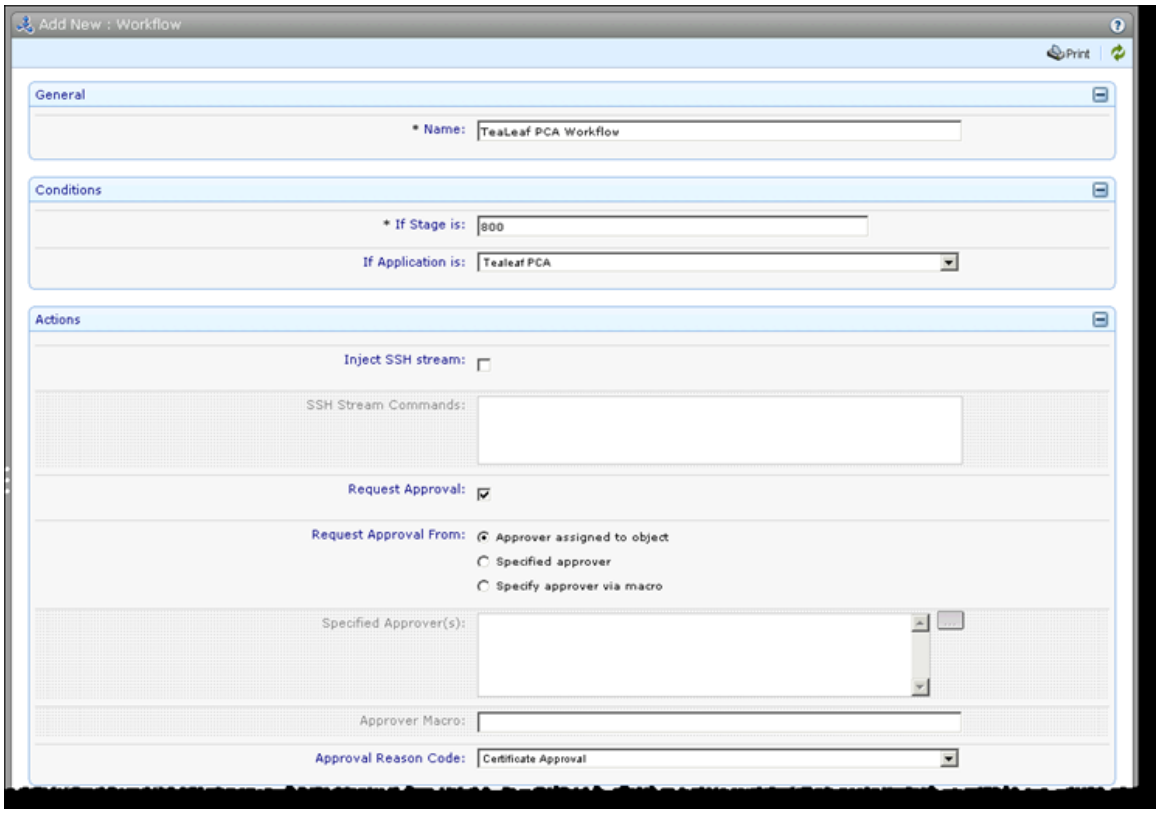

Alias objects are hidden by default. To view Alias objects in the Policy tree, you must enable the **Show Aliases** option. In the Windows console, click **File** > **Preferences > Policy Tree**, then select **Show Aliases**. In Policy Tree, click **Show All** > **Show Aliases**.

9. In Policy Tree, create workflow objects that require approval at stage 800 for the Tealeaf PCA application and all the web servers the Tealeaf PCA server is monitoring.

   For more information on defining Workflow objects, see "Protecting server platforms and keystores" on page 223.

   You must require approval at stage 800 for the Tealeaf PCA server and *all* the web servers it is monitoring so you can ensure that the web servers' certificates are provisioned to the Tealeaf PCA server *before* they are provisioned to their associated web servers. Otherwise, the Tealeaf PCA server monitoring functions will be disrupted.

   The Workflow object configuration for the Tealeaf PCA application will appear as follows:




You must create a corresponding Workflow object for each web server application that the Tealeaf PCA server monitors.

When you receive the certificate approvals for the Tealeaf PCA server and all its monitored web servers, approve the certificate installation on the Tealeaf PCA server first. After the private key is installed on the Tealeaf PCA server, you can approve the certificate installation on the Tealeaf PCA server's monitored web servers.

## Creating a Tealeaf PCA application object

To enable Trust Protection Foundation to automate the process of installing and maintaining your web servers' private keys on the Tealeaf PCA server, you must configure the Tealeaf PCA Application object. This object provides the information Trust Protection Foundation needs to manage your web server's private keys on the Tealeaf PCA Server.

**BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

### To create and configure a Tealeaf PCA application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **Tealeaf PCA**.
3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.
4. (Optional) In the **Device Certificate** box, click ... to select and associate a certificate with the new application.

   > **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

   For detailed information on associating a certificate with an application, see Associating a certificate with an application object.
5. Under **General**, do the following:
   a. In the **Application Name** field, type a name for the new application.
   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.
   c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

      Default system notifications are sent to the contact identities. The default contact is the master administrator.

> **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
>
> Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

   The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

   You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

   For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Under **Application Information**, do the following:

   a. Click  next to **Application Credential** to browse for the credential object that you want to use to authenticate with the application.

   > **DID YOU KNOW?** Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential might be a user name or private key credential; some drivers–such as F5, which is not SSH-based–can only use the user name credential for authentication.

   > **NOTE** The user account you select must have *Read* and *Write* access to the Temporary, Private Key, and Certificate directories.
   >
   > If you need help with this step, see your system administrator.
   >
   > For more information, see Working with system credentials in the *Administration Guide*.

> **DID YOU KNOW?** The **Connection Method** is the protocol that Trust Protection Foundation uses to connect to the server and manage the certificates installed on that server. In an application object's settings, this field is typically read-only.

b. Click the **Connection Method** list, click the protocol to use–HTTPS or SSH–and then in the **Port** field, specify the associated port number.

c. (Optional) In the **Port** field, type the port that Trust Protection Foundation should use to communicate with the server where the application is installed.

   Trust Protection Foundation uses the SSH protocol to communicate with the application server installed on Linux or Windows. The default SSH port assignment is port 22.

7. Complete the settings for the application object by referring to the following table:

| Field | Policy | Description |
|---|---|---|
| **Passive Capture** | | |
| Passive Capture Setup Path |  | The path to the directory where the Tealeaf PCA package is installed.<br>The default path is /usr/local/ctccap. |
| **Private Key Information** | | |
| Private Key File | No | The name of the private key currently installed in the Passive Capture Setup Path.<br>This option only needs to be configured if the private key currently in use was not installed by Trust Protection Foundation. Providing this value allows Trust Protection Foundation to remove the old private key after it is replaced. For more information, see Managing Tealeaf PCA Application Objects Via Policy.<br>If Trust Protection Foundation installed the current private key, it will automatically remove the key once it becomes the third generation key. |

8. When you are finished, click **Save**.

## What's next?

After you've created an application object, here are other things you can do to manage the new application:

- On the application's **Settings** sub-tab:
  - Click  to push a certificate to its associated application.

    For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.
  - Click  **Reset** to stop processing the application and reset the status and stage.
  - Click  to reattempt installation of the certificate to its associated application, .
  - Click  **Validate Now** to validate the applications associated certificate.

    Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

    For more information, see "About certificate and application validation" on page 270.
- On the application object's **Validation** tab, you can configure validation settings for the application object.
- On an object's **General** tab:
  - Click the **Log** sub-tab to view any events that are triggered by the template object.
  - Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

# VAM nShield configuration

Many organizations that provide online services use Agentless Monitoring Devices (AMD) to monitor web server response time. AMD passively intercepts incoming requests and determines the response time required to fulfill the request from the web server. To monitor encrypted traffic, the AMD leverages the web server certificate's private key which is stored on the AMD.

CyberArk Trust Protection Foundation™ automates the process of installing and maintaining your web server's private keys on the AMD. At present, Trust Protection Foundation supports the Compuware* Vantage* Agentless Monitoring Device (VAM) on Entrust nShield Connect XC HSM devices.

Trust Protection Foundation automatically installs the web server certificate's private key on the VAM device when you associate the VAM nShield Application object with the web servers' associated Certificate object.

To secure the transfer of private keys to the Vantage AMD device, Trust Protection Foundation encrypts the private key, then transfers the private key using a password-protected package. After it validates that the private key was successfully transferred, Trust Protection Foundation removes all temporary files.

This section provides the information required to provision web server certificate's private keys on VAM devices.

This chapter contains the following topics:

## VAM nShieldCertificate Lifecycle

The following table outlines the lifecycle stages for private keys installed on nShield NetHSM cards and the management level associated with each stage.

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| Stages 0-700 are performed by the X.509 Certificate Application driver on the Trust Protection Foundation server.<br><br>The VAM nShield application driver does not support remote key generation on VAM nShield devices. | | | | |
| 0 | StartProcessing | No processing. | | |
| 100 | CheckStore | No processing. | | |
| 200 | CreateConfigureStore | No processing. | | |
| 300 | CreateKey | No processing. | | |
| 400 | CreateCSR | No processing. | | |
| 500 | PostCSR | No processing. | | |
| 600 | ApproveRequest | No processing. | | |
| 700 | RetrieveCertificate | No processing. | | |
| 800 | InstallPrivateKey | Trust Protection Foundation installs the private key if the private key was not created on the host. | | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 801 | Roll Generations | Trust Protection Foundation removes the grandparent key, then rotates the current key.<br><br>For more information, see "Managing Key Rotation on nShield NetHSM Cards" on page 594. | | |
| 802 | Restart Agentless Monitoring Device | Trust Protection Foundation restarts the VAM Application.<br><br>Trust Protection Foundation only restarts the RTM device if the **Restart Device** option is enabled in the VAM nShield Application object. | | x |
| 900 | CheckConfiguration | This stage is reserved for future use. | | x |
| 1000 | ConfigureApplication | This stage is reserved for future use. | | x |

| Stage | Friendly Name | Description | Enrollment | Provisioning |
|---|---|---|---|---|
| 1100 | RestartApplication | This stage is reserved for future use. | | x |
| 1200 | EndProcessing | Trust Protection Foundation completes the certificate processing and, if configured, runs a Validation check on the keys. | | x |
| 1400 | Revocation | Trust Protection Foundation submits a revocation request to the CA.<br>Certificate revocation is a certificate operHation; it does not involve the application driver. | x | x |

## VAM Permission Requirements

When Trust Protection Foundation provisions private keys or conducts an onboard Validation on a VAM device, the user account that Trust Protection Foundation uses to authenticate with the device must have the following permissions:

- Sufficient permissions to access the private key stored in the designated module on the NetHSM card.
- Sufficient permissions on the AMD to run needed commands, such as generatekey, nfkminfo,pkcsmgr,rtm, and rcmd.
- Read and write access to the Temp Directory defined in the Device object. For more information on the Device object configuration, see "Creating a device object in the Policy Tree" on page 239.

## VAM nShieldPrerequisite Configuration

Trust Protection Foundation automates the process of installing and maintaining your web server's private keys on the VAM device. To allow Trust Protection Foundation to provision your web server certificate's private keys on VAM devices, you must configure Trust Protection Foundation.

However, before you configure Trust Protection Foundation, you must set the ssl.import.all.keys.from.token property on your AMD to *true*. Doing so overrides the settings specified in server.key.list and causes the AMD to read the keys from the accelerator card.

During these procedures, you must have either Write or both Associate and View permissions to the certificate object in order to select them. For more information, see About Permissions and Inheritance in the *Trust Protection FoundationAdministration Guide*.

### To edit the ssl.import.all.keys.from.token property

1. On your AMD, open the /usr/adlex/config/rtm.config configuration file.
2. Locate the ssl.import.all.keys.from.token property and set it to *true*.

   If this property does not exist in your rtm.config file, create it and set it to true.

### To configure Trust Protection Foundation to work with VAM

1. Open the SSH port to the VAM device.

   Trust Protection Foundation uses the Secure Shell (SSH) protocol to manage private keys on NetHSM cards; therefore, Trust Protection Foundation must have access to the device's SSH port. The default SSH port is port 22.
2. (Recommended) Create a Policy object for the VAM device and all its associated web servers and certificates.

   The VAM device and its associated web servers require a one-to-many configuration and Workflow objects specific to those applications. Consequently, it is easier to group all objects associated with the VAM configuration under a single Policy object.
3. In the Policy Tree, create a Device object for the machine that hosts VAM nShield NetHSM device.

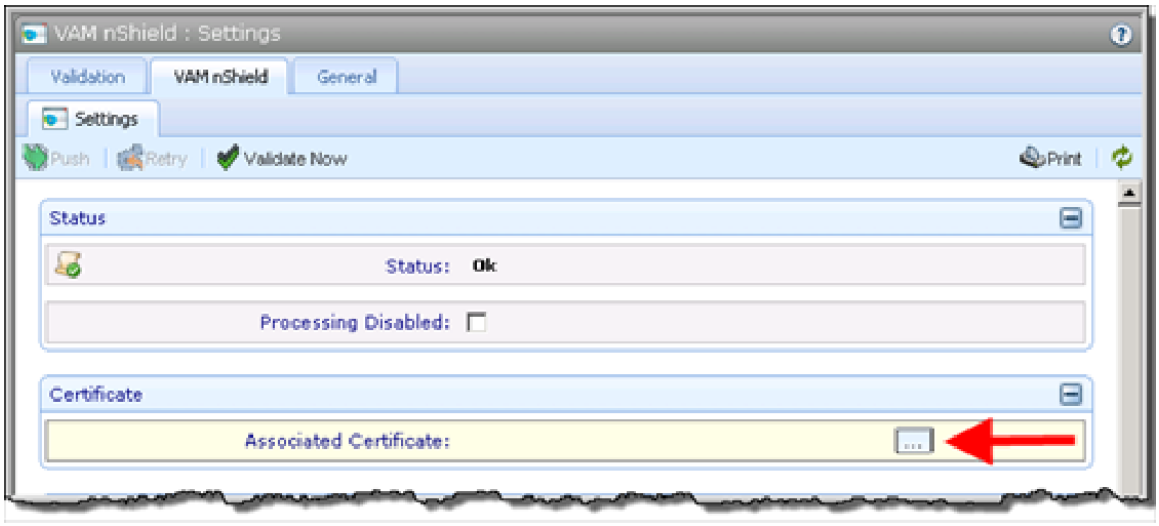
   For more information, see "Creating a device object in the Policy Tree" on page 239.
4. In the Policy Tree, create and configure an application object for the VAM nShield NetHSM device.

You must make sure that processing is not disabled in the VAM nShield Application object for Trust Protection Foundation to provision the web server certificate to the VAM device.

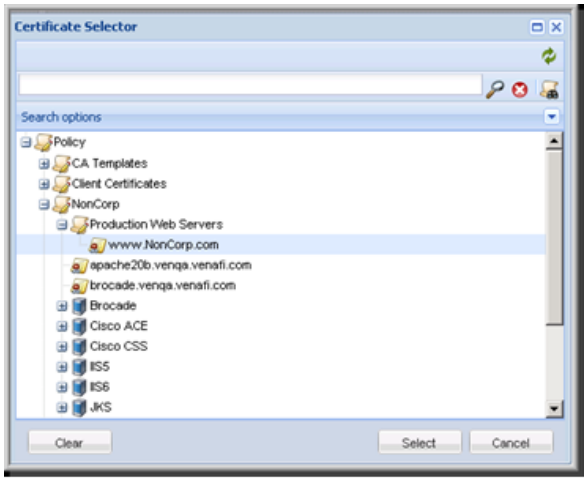
For more information, see "Creating a VAM nShield application object" on page 589.

For more information on creating application objects, see "Creating an application" on page 265. For details on the object's settings, see "Creating a VAM nShield application object" on page 589.

5. In the VAM nShield application object's **Associated Certificate** field, click the **Browse** button.

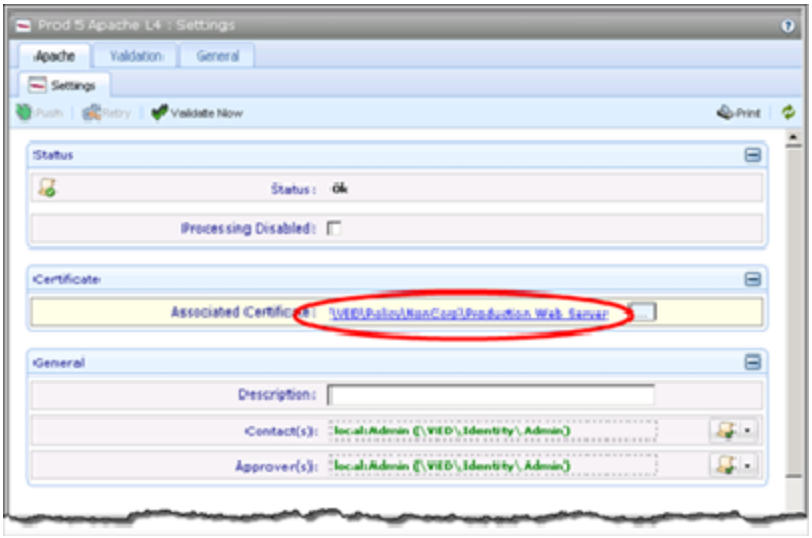



6. In the **Certificate Selector** dialog, select the certificate installed on a web server that the VAM is monitoring.




7. Click **Select**.

The certificate object is associated with the current application.



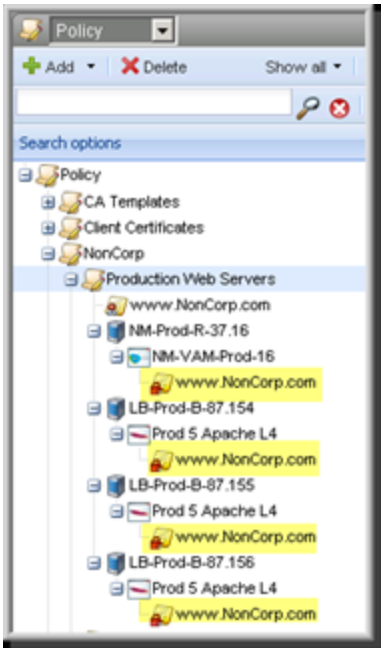

8. Repeat steps 4-7"Click Select." on the previous page for every certificate installed on a web server that the VAM is monitoring.

   Unlike most applications managed by Trust Protection Foundation, the VAM nShield application references the certificate being hosted by a web server or group of web servers in order to decrypt traffic destined for that server. Each application object references a single certificate, so if you have multiple SSL certificate private keys that are needed for decryption purposes, you will need multiple application objects.

   When you associate the VAM nShield application object with the web server certificate, the associated certificate object is listed in the VAM nShield application object as follows:

Additionally, if enabled, an alias object for the web server certificate associated with the VAM device appears under the VAM nShield application object.

The certificate alias object also appears under the web server application objects where the certificate is installed.



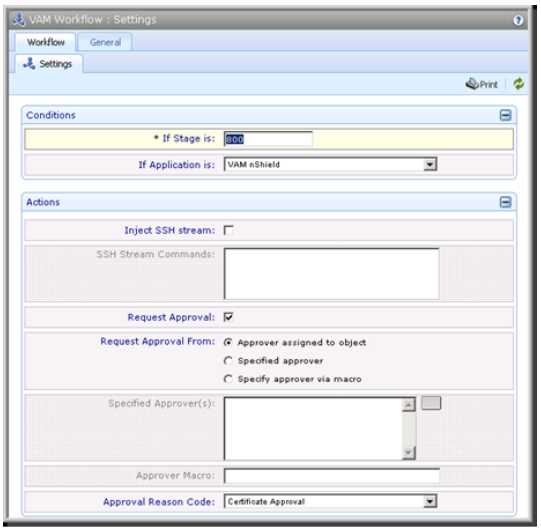

> **TIP** Alias objects are hidden by default. To view Alias objects in the Policy tree, you must enable the **Show Aliases** option. In the Windows console, click **File** > **Preferences > Policy Tree**, then select **Show Aliases**. In Policy Tree, click **Show All** > **Show Aliases**.

9. In the Policy Tree, create Workflow objects that require approval at stage 800 for the VAM nShield application and all the web servers the VAM device is monitoring.

   For more information on defining Workflow objects, see "Protecting server platforms and keystores" on page 223.

   You must require approval at stage 800 for the VAM device and *all* the web servers it is monitoring so you can ensure that the web server's certificates are provisioned to the VAM device *before* they are provisioned to their associated web servers. Otherwise, the VAM monitoring functions will be disrupted.

   The Workflow object configuration for the VAM nShield application will appear as follows:




Keep the following points in mind:

- You must create a corresponding Workflow object for each web server application that the VAM device monitors.
- When you receive the certificate approvals for the VAM device and all its monitored web servers, approve the certificate installation on the VAM device first.
- After the private key is installed on the VAM device, you can approve the certificate installation on the VAM's monitored web servers.

## Creating a VAM nShield application object

To enable Trust Protection Foundation to automate the process of installing and maintaining your web servers' private keys on the AMD, you must configure the VAM nShield Application object. This object provides the information Trust Protection Foundation needs to manage your web servers' private keys on the AMD.

**BEST PRACTICE** Consider managing object settings using a policy. For more information, see "Managing applications using policies" on page 269.

### To create and configure a VAM nShield application object

1. From the Certificate Manager - Self-Hosted menu bar, click Policy tree.
2. In the **Policy** tree, select the device object to which you want to add the new application object, and then click **Add** > **Application**, and then select **VAM nShield**.
3. When the new application object page appears, then under **Status**, clear the **Processing Disabled** checkbox.

   When checked, this option disables provisioning of the certificates installed on the current application. This means that Trust Protection Foundation does not attempt to install, renew, process, or validate certificates on the application.
4. (Optional) In the **Device Certificate** box, click ... to select and associate a certificate with the new application.

   > **NOTE** If you don't have a certificate ready, you can do this later or you can do it on the certificate's **Association** tab.

   To associate a certificate with the current application, you must have *write* permissions to the application object and either *write* or *associate* permissions to the certificate object.

   For detailed information on associating a certificate with an application, see Associating a certificate with an application object.
5. Under **General**, do the following:
   a. In the **Application Name** field, type a name for the new application.
   b. (Optional) In the **Description** field, type a description for the purpose of the application.

      A strong description can help to provide context for other administrators who might need to manage the new application.
   c. In the **Contacts** field, select user or group identities you want assigned to this application object (or choose the Use policy value to configure contacts using a policy).

      Default system notifications are sent to the contact identities. The default contact is the master administrator.

> **TIP** If the Identity Selector dialog is not populated when it first opens, enter a search query to retrieve the Identity list. The administration console does not automatically display external users and groups. You must first enter a search string so Trust Protection Foundation can query the external Identity store, then return the list of requested users or groups. If you want to display all user or group entries, enter the wildcard character (*).
>
> Press **Shift+click** to select multiple, contiguous users and groups. Press **Ctrl+click** to select multiple, discontiguous users and groups.

d. In the **Approvers** field, select user or group Identities you want to assign to approve workflows (certificate approval or injection command) for the new application.

   The default approver is the master administrator. For more information on defining workflow objects, see Implementing certificate workflow management.

e. (Conditional) If your application (or certificate) object is affected by a defined workflow and you want users to use a console other than Policy Tree, click **Managed By** and select which administration console to use as part of the workflow.

   You only need to configure this if you are using workflows and expect users to perform a task using a particular administration console. The default setting is Policy Tree.

   For more information, see Specify folders and certificates to be managed by Certificate Manager - Self-Hosted .

6. Under **Application Information**, do the following:

   a. Click  next to **Application Credential** to browse for the credential object that you want to use to authenticate with the application.

   > **DID YOU KNOW?** Credential objects store the credentials Trust Protection Foundation uses to authenticate with devices, applications, and CAs. The stored credential might be a user name or private key credential; some drivers–such as F5, which is not SSH-based–can only use the user name credential for authentication.

   > **NOTE** The user account you select must have *Read* and *Write* access to the Temporary, Private Key, and Certificate directories.
   >
   > If you need help with this step, see your system administrator.
   >
   > For more information, see Working with system credentials in the *Administration Guide*.

> **DID YOU KNOW?** The **Connection Method** is the protocol that Trust Protection Foundation uses to connect to the server and manage the certificates installed on that server. In an application object's settings, this field is typically read-only.

b. Click the **Connection Method** list, click the protocol to use–HTTPS or SSH–and then in the **Port** field, specify the associated port number.

c. (Optional) In the **Port** field, type the port that Trust Protection Foundation should use to communicate with the server where the application is installed.

   Trust Protection Foundation uses the SSH protocol to communicate with the application server installed on Linux or Windows. The default SSH port assignment is port 22.

7. Complete the settings for the application object by referring to the following table:

| Field | Policy | Description |
|---|---|---|
| **Base Settings** | | |
| nShield Setup Path |  | The path to the utilities that the VAM nShield application driver uses to install the certificate and private key on the NetHSM card.<br><br>**NOTE** This is the bin Directory where the nFast installation was performed. The default path is /opt/nfast. |
| Module ID |  | The Module ID of the partition where the private key is stored on the NetHSM card.<br><br>The Module ID is the first module number returned. On NetHSM devices, the first Module ID is 1. |
| Restart Device |  | Select **Yes** if you would like Trust Protection Foundation to automatically restart the RTM device after it installs the private key. |
| **Private Key Settings** | | |
| Private Key Label | No | The name of the private key currently installed on the NetHSM card.<br><br>**IMPORTANT** This option only needs to be configured if the private key currently in use was not installed by Trust Protection Foundation. Providing this value allows Trust Protection Foundation to remove the old private key after it is replaced. For more information, see "Managing Key Rotation on nShield NetHSM Cards" on the next page.<br><br>If Trust Protection Foundation installed the current private key, it will automatically remove the key once it becomes the third generation key to preserve space on the NetHSM card. |

8. When you are finished, click **Save**.

## What's next?

After you've created an application object, here are other things you can do to manage the new application:

- On the application's **Settings** sub-tab:
  - Click  to push a certificate to its associated application.

    For more information, see Pushing a certificate and private key to an application in the *Certificate Management Guide*.
  - Click  **Reset** to stop processing the application and reset the status and stage.
  - Click  to reattempt installation of the certificate to its associated application, .
  - Click  **Validate Now** to validate the applications associated certificate.

    Validation requests are placed into a queue. When your validation runs, the application and its associated certificate are scanned according to the settings configured in the application object's **Validation** tab.

    For more information, see "About certificate and application validation" on page 270.
- On the application object's **Validation** tab, you can configure validation settings for the application object.
- On an object's **General** tab:
  - Click the **Log** sub-tab to view any events that are triggered by the template object.
  - Click the **Permissions** sub-tab to configure the users or groups to whom you want to grant permissions to the new object. For more information, see Permissions overview in the *Trust Protection FoundationAdministration Guide*.

## Managing Key Rotation on nShield NetHSM Cards

The VAM nShield application driver uses a Generational credential to manage generations of private keys. The Generational Credential object is a driver-generated credential that stores the credentials required to remove a certificate and/or private key after it is replaced. There are no configurable options in a Generational Credential object. It is a transient credential that is updated every time Trust Protection Foundation renews a certificate and/or private key.

The Generational credential stores three generations of private key credentials. These credentials can only be edited in their first generation; that is, the credential required to access the private key currently installed on the NetHSM card.

If you want Trust Protection Foundation to remove the current private key after it is replaced (the key will only be removed when it is in its third generation), you can edit the first generation credential in the VAM nShield Application object. Providing this value allows Trust Protection Foundation to remove the old private key after it is replaced.

> **NOTE** If Trust Protection Foundation installed the current private key, it will automatically remove the key once it becomes the third generation key to preserve space on the NetHSM card.

### To configure the credential for the private key currently installed on the NetHSM card

1. From the SSH Manager for Machines menu, click **Configuration** > **Policy Tree**.

   You must have Read and Write permissions to the VAM nShield Application object.

2. In the Policy tree, select the VAM nShield Application object.

3. Click **Set/Change**.

4. Enter the name of the private key currently installed on the NetHSM card.

   Providing this value allows Trust Protection Foundation to remove the old private key after it is replaced.

   This option only needs to be configured if the private key currently in use was not installed by Trust Protection Foundation. If Trust Protection Foundation provisioned the current private key, it automatically removes the private key once it becomes the third generation key to preserve space on the NetHSM card.

5. Click **Save/Apply** to save your changes.

6. Once this key is the third generation, Trust Protection Foundation will remove the private key.

# A

## B

## C

## D

## E



## F



## G

## H



## I

## J

## K

## L



## M



## N

## O



## P

## Q



## R

## S

## T



## U



## V

## W